Gaze Toolset

v3.5.1

Generated by Doxygen 1.8.17

Chapter 1

Changelog

v3.5.0

New Features

- Allow to configure the `BackgroundColor` and the `FrameColor` of the windows (#8).
- Calibration accuracy
  - Estimate the calibration accuracy by capturing gaze data during the calibration process and associating the captured data to the calibration result.
  - Add the calibration accuracy values to the calibration data output (#6).
  - Show an alert in the calibration result view if the calibration accuracy is poor. The threshold can be configured with `CalibrationAccuracyThreshold` (#6).
  - Allow to configure automatic calibration restart with the configuration item `CalibrationRetries` (#6).
- Validation accuracy and precision
  - Show an alert in the validation result view if the validation accuracy or precision is poor. The thresholds can be configured with `ValidationAccuracyThreshold` and `ValidationPrecision↩Threshold`, respecvely (#6).
  - Allow to configure automatic validation restart with the configuration item `ValidationRetries` (#6).
- Improve calibration result view:
  - Don't display gaze points for failed calibration points.
  - Color failed calibration points red in calibration result view.
  - Draw fine lines from gaze points to the corresponding calibration points to allow for an easy association.
- `GazeControl.exe` returns now an error code which allows to check whether the command was executed correctly.
- Add a systray icon. This allows to control the application without having to use the `GazeControl` application.

Improvements

- Place log files in folder `log` (#2).
- Add log entries for pipe server (#3).
- Skip failed calibration point and continue calibration.
- Check and sanitize calibration and validation points to avoid setting the same point multiple times.

Changes

- Changes to the calibration output. This requires a new value for the configuration items `Calibration↩ LogColumnTitle` and (optionally) `CalibrationLogColumnOrder`.

Bug Fixes

- Fix keyboard focus when a window is opened (#11).
- Fix validation data output (y values of the validation points).

v3.4.2

Improvements

- Seperate pipe command handler funtionality to `NamedPipeClient` class and move `NamedPipeClient` to seperate project based on .NET 4.72. This allows for a direct integration in python with `pythonnet`.

Bug Fixes

- Log GazeControl messages to seperate file to avoid race conditions (#1).
- Fix pipe connection problems on startup (#5).
- Only hide the mouse when calibration dots are shown (#7).
- Fix dispersion calculation in drift compensation (#9).

v3.4.1

Improvements

- Add note on foreground behaviour to readme.

Bug Fixes

- Fix validation when only one screen is connected.

v3.4.0

New Features

- Add configuration option `DriftCompensationWindowShow` to enable or disable the drift compensation window.
- Add configuration option `DriftCompensationDurationThreshold` to configure the required fixation time during drift compensation.
- Add configuration option `DriftCompensationWindowShow` to enable or disable the drift compensation window.
- Add configuration option `DriftCompensationDispersionThresholdMax` which allows to define maximal allowed deviation angle during drift compensation: If the computed compensation angle is larger as the here configured angle the drift compensation is not updated.
- During `Gaze.exe` startup check for already running `Gaze` processes and kill them.

Improvements

- Update default configuration settings to allow starting the application without error.
- Improvments to `GazeControl.exe` and annotations:
  - Introduce arguments `/label` and `/trialId` to pass a label and the trial ID.
  - Make argument `/command` optional to allow setting annotations without a command.
  - Set trialID and label based on capture timestamp instead of system timestamp.
- Improve timestamps: Compute the tracker latency for each sample and use this to dump the system time of data capure and the system time of data reading.
- Update opensesame templates. This might help to reduce taskbar flickering and makes the code more maintainable.

Changes

- Remove argument `/value` in `GazeControl.exe` because it became obsolete with the new arguments `/label` and `trialId`.
- Rename configuration option `DispersionThreshold` to `DriftCompensationDispersion↩ Threshold`.
- Hide window icon in taskbar. This might help to avoid taskbar flickering during experimentatino.
- Save data files with `csv` extension instead of `txt`.
- By default, use `,` delimiter instead of `\t` delimiter when dumping values.

Bug Fixes

- Fix dispersion computation during drift compensation.

v3.3.2

New Features

- Allow to select the screen on a multi-screen setup with the keyboard.

Improvements

- Improve opensesame template files and add a template for version 3.3 and 4.0.
- Improve ztree template file
- Add a documentation to the sample folder

Bug Fixes

- Move calibration, validation and drift compensation windows to the foreground.

v3.3.1

Improvements

- Improve performance.
- Dump validation results for each validation point.

Bug Fixes

- Represent the relative timestamp in total milliseconds instead of a timespan.

v3.3.0

New Features

- Add pipe command `SET_TRIAL_ID` to allow to annotate data samples.
- On multi-screen setups start calibration with a screen selection page.
- Add Screen Area coordinates to the dumped configuration file.
- Add gaze validation which can be started through the command `VALIDATE`.

Improvements

- Fix compiler warnings.
- Extend helper scripts.

v3.2.0

New Features

- Add relative timestamp to output data.
- Add annotation tag to ouput data.
- Add pipe command `SET_TAG` to allow annotate data samples.
- Add pipe command `RESET_START_TIMER` to reste the relative timestamp.
- Add a log entry of the version of the gaze application.
- Add helper scripts to generate shortcuts to `GazeControl.exe`.

v3.1.0

New Features

- Add a custom drift compensation process
- Allow to pass the argument `outputPath` to the application for dynamic output path assignement.

Improvements

- Integrate calibration into `Gaze.exe`
- Remove Tobii research dependencies from everywhere except the eye tracker device class
- Cleanup and rearrangement of code to improve readability

v3.0.0

New Features

- A custom calibration application is added to the portfolio. This allows to calibrate a device without the need for a 3rd party application.
- Proper shutdown handling of `GazeToMouse` through named pipes.
- Allow to enable/disable gaze recording through named pipes.
- Allow to enable/disable mouse tracking through named pipes.
- Allow to pass argument subject to the application.

Improvements

- Update all projects to .NET version 6.0.
- Cleanup code base, split functions into seperate libraries.
- Apply MVVM architectural pattern where sensible.

Changes

- Remove Tobii Interaction Library
- Remove all configuration options for Tobii Core (only Tobii Pro SDK is supported)
- Remove Tobii Core application wrapper (TobiiTest, TobiiGuestCalibrate)
- Use the Tobii pro eye tracker manager for device calibration instead of the Tobii Core software.
- Rename `GazeToMouse` to `Gaze` and `GazeToMouseClose` to `GazeClose`.

v2.3.0

New Features

- A mouse tracker device can now be used instead of an eyetracker device. The mouse tracker logs the timestamp and the x and y coordinates of the mouse pointer whenever the mouse-move event is fired. The mouse tracker is used when the configuration filed 'TrackerDevice' is set to the value 2.

Improvements

- Rename the configuration field 'TobiiSDK' to 'TrackerDevice'.

v2.2.0

New Features

- Configuration file
  - Dump the configurations used for an experiment to a file at the "DataLogPath"
  - Allow to configure an experiment name which is used as a postfix of the dumped configuration file name
  - Consider the config file as invalid if not all required configuration parameters are defined
  - Consider the config file as invalid if unknown parameters are defined
  - Allow to configure whether to log data sets where all data is invalid (eyes closed, no subject in front of the screen, etc)
- Error Handling
  - Attach an error string to the output file, indicating errors that occurred during the run
  - Attach an error string to the dumped configuration file, indicating errors of the configuration

Improvements

- Fall back to Core SDK if the license file cannot be applied to the device

v2.1.0

New Features

- Log eye origin coordinates
  - x, y, z coordinates of the left and the right eye
  - compute distance of the left and right eye to the eyetracker
  - compute the average distance of the two eyes

Improvements

- Check the three format values and the column order individually to produce more specific log entries

v2.0.1

Bug Fix

- with SDK Pro, use system timestamp to cope with disconnected device
- fix the path in the z-tree sample file

v2.0.0

New Features

- Support for Tobii Pro SDK
  - apply license to eyetracker device at stratup
  - logging of pupil diameter
  - logging of individual eye data
- Allow to configure column headers of output file

Improvements

- Improved configuration options for the output file

v1.0.0

New Features

- Notify user with popup if eyetracker is not ready
- Allow to configure time interval for the software to wait for the eyetracker to become ready

Improvements

- Rename default output file for data from `<prefix>_data.txt` to `<prefix>_gaze.txt`

v0.3.2

Improvements

- add header to the data log file.
- change the default value of allowed gaze data files.
- check and wait for ready state of the eye tracker before performing operations with it.

Bug Fix

- create a log file per machine to prevent concurrency conflicts.

v0.3.1

Improvements

- ignore the option "HideMouse" when "ControlMouse" is disabled.

Bug Fix

- remove double log entry of mouse hiding and restoring event.

v0.3.0

New Features

- allow to configure whether the gaze data is logged.
- allow to configure the maximum allowed amount of gaze data files in the output folder. Oldest files are deleted first.

Improvements

- limit the logfile size to 1MB. If the size is exceeded a new file is created. At any time only two log files are allowed, The older file is overwritten once both files exceed 1MB.

v0.2.0

New Features

- allow to configure whether the mouse is controlled by the gaze of the subject or not.
- allow to configure the output format of the gaze data.

v0.1.0

First release of the GazeToMouse toolset.

The toolset was tested on **Windows 7** in conjunction with **ztree v3.6.7** and Tobii **Eye Tracking Core v2.11.1.6952**.

Chapter 2

Toolset to Control Tobii Eye Tracker

This repository contains the source code for multiple simple tools that allow to control a Tobii eye tracker from a 3rd party application. Specifically, this project aims at providing a set of executables that can be called from within `ztree` to allow eye tracker support for economic experiments.

For more details please refer to the `documentation`.

Important Configure the task manager to be always in the foreground (In task manager enable "Options->↩ Always on top"). Why: The application `Gaze.exe` may open windows that are put to the foreground in a very aggressive manner. This is done in order to cope with experimentation software that uses this same behaviour (e.g. Opensesame with psychopy or expyriment backend). If something goes wrong with `Gaze.exe` the user could be locked out from the computer because a window keeps blocking access to the system. With the task manager set to "Always on top" there is a way out.

Installation

The complete toolset package can be downloaded from the `releases`. The package contains the following executables:

- `**Gaze.exe**` This program uses the `Tobii Pro SDK` to extract the gaze position on the screen where the subject is looking at. The extracted data is recorded and stored to a file. Optionally, the mouse cursor position is updated to this position such that the mouse cursor is controlled by the gaze of the subject. Instead of using an eye tracker device it is also possible to simply log the mouse coordinates. `**Gaze.↩ exe**` runs infinitely until it is terminated by an external command. This should **not** be done with a forced kill (e.g. by executing the command `taskkill /F /IM Gaze.exe` or by killing the task with the task manager) because it prevents the program from terminating gracefully. This as several consequences:
  - open files are not closed properly and the data stream is cut off. This can lead to corrupt files.
  - if the feature of hiding the mouse pointer is used, the mouse will remain hidden.
  - memory is not freed properly. Instead the program `**GazeControls.exe /command TERMI↩ NATE**` should be used.
- `**GazeControl.exe**` This program allows to interact with `**Gaze.exe**`. `GazeControl.exe` accepts the following optional arguments:
  - `/reset`: Allows to reset the relative timestamp of the gaze data.
  - `/trialId <ID>`: Sets a trial ID `<ID>` which will be added to each data sample in the output file. **Important**: Make sure that only integer numbers are used as trial ID.

  - /label <LABEL>: Sets a custom label <LABEL> which will be added to each data sample in the output file. Any string is accepted here.
  - /command <COMMAND>: A command allow to activate/deactivate features of Gaze.exe. The following commands are supported:
    * CUSTOM_CALIBRATE uses the Tobii Pro SDK and launches a custom calibration process which allows to calibrate the eye tracker without having to rely on the calibration software provided by Tobii.
    * DRIFT_COMPENSATION launches a custom drift compensation process to compensate gaze drifts that may occur during experimentation.
    * GAZE_RECORDING_DISABLE requests **Gaze.exe** to stop recording gaze data. Gaze.exe will continue to run (and update the mouse pointer if configured accordingly) but no longer store gaze data to the disk.
    * GAZE_RECORDING_ENABLE requests **Gaze.exe** to start recording gaze data.
    * LOAD shows a window with a spinner for a configurable amount of time.
    * MOUSE_TRACKING_DISABLE requests **Gaze.exe** to stop updating the mouse pointer by the gaze position.
    * MOUSE_TRACKING_ENABLE requests **Gaze.exe** to start updating the mouse pointer by the gaze position.
    * RESET_DRIFT_COMPENSATION resets the drift compensation computed with the command DRIFT_COMPENSATION.
    * TERMINATE requests **Gaze.exe** to close gracefully and logs these events to the log file.
    * VALIDATE uses the Tobii Pro SDK Addon and launches a validation process.

  Multiple arguments can be passed to the application but each argument can only be passed once. Passing an argument to an application can be done in command line or by crating a shortcut to the program. Corresponding shortcuts for all available <COMMAND>s are provided in the release package.

- **ShowMouse.exe** This program allows to restore the standard mouse pointer. It might be useful if the program Gaze.exe crashes or is closed forcefully such that the mouse pointer is not restored after terminating. The subject might end up with a hidden mouse pointer. A good solution for such a case is to install a shortcut to ShowMouse.exe on the desktop in order to execute it with the keyboard.

In order to run the executables the following files need to be placed in the same directory as the executables:

- tobii_pro.dll
- tobii_firmware_upgrade.dll
- assets/blank.cur
- config.json

In order to use tje GazeControlLibrary, the following files need to be placed in the same directory as the executables:

- GazeControlLibrary.dll
- Newtonsoft.Json.dll

Further, the Tobii engine must be running and the eye tracker must be enabled.

Tobii Eye Tracker 4c

To install the driver for the `Tobii Eye Tracker 4c` install `Tobii Experience Driver`.

This will start the following services:

- `Tobii Runtime Service`
- `Tobii Service`

and the following processs:

- `Tobii Interaction Engine`

Tobii Pro Spark

To install the driver for the `Tobii Pro Spark` use the `Tobii Pro Eye Tracker Manager`:

1. Install Tobii Pro Eye Tracker Manager (ETM)
2. Connect the Tobii Pro SPark device to the computer
3. Install the driver with the ETM

This starts the service `Tobii Pro Spark Runtime`.

Scripts

The folder `scripts` contains two files `CreateShortcut.ps1` and `CreateShortcuts.bat` which allow to create shortcuts to the application `GazeControl.exe` with predefined command arguments. In order to generate the shortcut files perform the following steps:

1. copy the two script files into the installation folder

1. execute the script `CreateShortcuts.bat`

Nothe that the generated shortcuts are tied to the installation folder. Copying the installation folder to another location will break the links.

3rd Party Applications

This section provides some infromation on how to run the here provided executables from within 3rd party applications.

ztree

For quick starters, a simple `ztree sample program` is available.

Opensesame

For quick starters, a simple `opensesame sample program` is available.

Release Notes

Information about the releases can be found in the `CHANGELOG`

Chapter 3

Sample Files for Experimentation with Eye Tracker Utility

This folder holds some sample files to use the gaze utility in an experiment management tool.

`config.json`

A sample configuration file which can be used as a starting point to configure the gaze utility.

`config_libgac.json`

A sample configuration file which produces minimalistic gaze output data which matches with the example script of `libgac` (a gaze analysis library).

`template.osexp`

A sample file which demonstrates how to start the gaze utility from openseame. This was tested on opensesame version `3.3.14` and `4.0.5` on Windows.

Note that the application only worked with the PyGame (legacy) backend because otherwise the gaze windows kept beeing covered by the opensesame fullscreen window.

It might be possible (and potentially a better solution) to manually control the window through python (e.g. with win32gui on Windows or with xdotool on Linux).

`template.ztt`

A sample file which demonstrates how to start the gaze utility from openseame. The sample file was generated with the ztree version `5.1.11`.

Chapter 4

Namespace Index

4.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 6

Class Index

6.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 7

Namespace Documentation

7.1 CustomCalibrationLibrary Namespace Reference

7.2 CustomCalibrationLibrary.Commands Namespace Reference

Classes

- class CalibrationCommand

  Comand class to trigger calibration events.
- class **GazeVisibilityCommand**

  Command class to change the gaze visibility
- class **ScreenSwitchCommand**

7.3 CustomCalibrationLibrary.Converters Namespace Reference

Classes

- class NotBoolVisibilityConverter

  Converts True to Hidden and False to Visible
- class PositionConverter

  Converter class to convert a normalized coordinate to a pixel coordinate.
- class ProximityColorConverter

  Converter class to convert the proximito of a normailezed coordinate to the center point (0.5) into colors.

7.4 CustomCalibrationLibrary.Extensions Namespace Reference

Classes

- class BrushExtension

  Allows to attach a brush property.

7.5 CustomCalibrationLibrary.Models Namespace Reference

Classes

- class CalibrationModel

  The model for the calibration process.

Enumerations

- enum CalibrationEventType {
  Init, **Start**, **Accept**, **Restart**,
  Abort }

  Events to trigger changes in the calibration process.
- enum CalibrationStatus {
  ScreenSelection, **HeadPosition**, **DataCollection**, **Computing**,
  CalibrationResult, **ValidationResult**, **Error**, **Disconnect** }

  The status of the calibarion process.

7.5.1 Enumeration Type Documentation

7.5.1.1 CalibrationEventType

`enum CustomCalibrationLibrary.Models.CalibrationEventType [strong]`

Events to trigger changes in the calibration process.

7.5.1.2 CalibrationStatus

`enum CustomCalibrationLibrary.Models.CalibrationStatus [strong]`

The status of the calibarion process.

7.6 CustomCalibrationLibrary.ViewModels Namespace Reference

Classes

- class CalibrationFailedViewModel

  The view model class for a failed calibration.
- class CalibrationPointViewModel

  The view model for a calibration point.
- class CalibrationResultViewModel

  View model class of the gaze calibration result.
- class CalibrationViewModel

  The view model class of the calibration view
- class ColoredViewModel

  The base view model for coloring the view.
- class DisconnectViewModel

  The view model class of the diconnect view
- class DriftCompensationViewModel

  The view model class of the drift compensation view.
- class Monitor

  A representation of the screen.
- class ScreenSelectionViewModel

  The view model class for the screen selection view.
- class UserPositionGuideViewModel

  The view model class for the user position guide view.
- class ValidationResultViewModel

  View model class of the gaze validation result.

7.7 CustomCalibrationLibrary.Views Namespace Reference

Classes

- class Calibration

  Interaction logic for Calibration.xaml
- class CalibrationFailed

  Interaction logic for CalibrationFailed.xaml
- class CalibrationFrame

  Interaction logic for CalibrationCollection.xaml
- class CalibrationPoint

  Interaction logic for CalibrationPoint.xaml
- class CalibrationResult

  Interaction logic for CalibrationResult.xaml
- class CalibrationResultLine

  Interaction logic for CalibrationResultLine.xaml
- class CalibrationResultPoint

  Interaction logic for CalibrationResultPoint.xaml
- class CalibrationWindow

  Interaction logic for MainWindow.xaml
- class Disconnect

  Interaction logic for Disconnect.xaml

- class DriftCompensationWindow

  Interaction logic for DriftCompensation.xaml
- class FixationPoint

  Interaction logic for FixationPoint.xaml
- class ScreenSelection

  Interaction logic for ScreenSelection.xaml
- class Spinner

  Interaction logic for Computing.xaml
- class UserPositionGuide

  Interaction logic for UserPositionGuide.xaml
- class ValidationResult

  Interaction logic for ValidationResult.xaml

7.8 GazeControl Namespace Reference

Classes

- class App

  Interaction logic for App.xaml

7.9 GazeControlLibrary Namespace Reference

Classes

- class **NamedPipeClient**

  The named pipe client handler.
- class PipeCommand

  The JSON structure of a pipe command.

Enumerations

- enum LogLevel { **Debug**, **Info**, **Warning**, **Error** }

  Defines the availabel log levels.
- enum ErrorCode {
  ErrorCode_Success = 0, **ErrorCode_ConnectionFailed**, **ErrorCode_RequestFailed**, **ErrorCode_No↩ Reply**,
  ErrorCode_Unknown }

  The error codes of the command handler.

7.9.1 Enumeration Type Documentation

7.9.1.1 ErrorCode

`enum GazeControlLibrary.ErrorCode [strong]`

The error codes of the command handler.

7.9.1.2 LogLevel

`enum GazeControlLibrary.LogLevel [strong]`

Defines the availabel log levels.

7.10 GazeToMouse Namespace Reference

Classes

- class App

  Interaction logic for App.xaml

7.11 GazeToMouse.Commands Namespace Reference

Classes

- class ExitApplicationCommand

  Command class to exit the application.
- class ResetDriftCompensationCommand

  Command class to reset the drift compensation.
- class StartCalibrationCommand

  Command class start the calibration.
- class StartDriftCompensationCommand

  Command class to start the drift compensation.
- class StartValidationCommand

  Command class to start the validation.
- class UpdateDriftDeviationAngleCommand

  Command class to start the drift compensation.

7.12 GazeUtilityLibrary Namespace Reference

helper class to show and hide the system curser

Classes

- class CalibrationDataError

  The calibration data error class to convert error flags to binary strings.
- class **ConfigChecker**

  Helper class to check for the valididty of configuration options.
- class ConfigItem

  configuration file class
- class ConfigScreenArea

  The JSON structure of the screen area.
- class DriftCompensation

  The class to handle drift compensation.
- class GazeConfigError

  The gaze config error class to convert error flags to binary strings.
- class GazeConfiguration

  The gaze configuration handler.
- class GazeDataError

  The gaze data error class to convert error flags to binary strings.
- class GazeError

  The base error class to convert error flags to binary strings.
- class JsonConfigParser

  The config file "config.json" is parsed and its values are attributed to the ConfigItem class.
- class MouseHider

  hide standard mouse pointer and resore it
- class ScreenArea

  The class describing the Screen area in 3d and 2d space.
- class ScreenTriangle

  A class to describe a triangle. This was supposed to be used to construct the ScreenArea but it turned out that it is simpler to work with the screen plane and use the normalised intersection points to check wheter the gaze point is outside the screen area.
- class TrackerLogger

  Simple logger class.
- class TrackerMessageBox

  Interaction logic for TrackerMessageBox.xaml

Enumerations

- enum EOutputType { **gaze**, **calibration**, **validation**, **control** }

  A list of output files.
- enum EGazeConfigError {
  FallbackToDefaultConfigName = 0x001, **FallbackToCurrentOutputDir** = 0x002, **FallbackToDefault↩ Config** = 0x004, **FallbackToDefaultDiameterFormat** = 0x008,
  FallbackToDefaultOriginFormat = 0x010, **FallbackToDefaultTimestampFormat** = 0x020, **OmitColumn↩ Titles** = 0x040, **FallbackToDefualtColumnOrder** = 0x080,
  FallbackToDefaultNormalizedFormat = 0x100 }

  Error values of the configuration
- enum EGazeDataError { **FallbackToMouse** = 0x01, **DeviceInterrupt** = 0x02 }

  Error values of the gaze output data
- enum ECalibrationDataError { **DeviceNotSupported** = 0x01, **DeviceInterrupt** = 0x02 }

  Error values of the gaze output data

7.12.1 Detailed Description

helper class to show and hide the system curser

7.12.2 Enumeration Type Documentation

7.12.2.1 ECalibrationDataError

enum GazeUtilityLibrary.ECalibrationDataError [strong]

Error values of the gaze output data

7.12.2.2 EGazeConfigError

enum GazeUtilityLibrary.EGazeConfigError [strong]

Error values of the configuration

7.12.2.3 EGazeDataError

enum GazeUtilityLibrary.EGazeDataError [strong]

Error values of the gaze output data

7.12.2.4 EOutputType

enum GazeUtilityLibrary.EOutputType [strong]

A list of output files.

7.13 GazeUtilityLibrary.DataStructs Namespace Reference

Classes

- class CalibrationPoint

  A calibration point class holding several metrics connected to a calibration point.
- class DriftCompensationData

  The drift compensation data structure
- class EyeData

  The eye data set, including pupil information.
- class GazeCalibrationData

  The gaze calibration data structure
- class GazeData

  The class definition of a gaze data set
- class GazeData2d

  The 2d gaze data set.
- class GazeData3d

  The 3d gaze data set.
- class GazeDataCollection

  The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data.
- class **GazeDataConverter**

  Convert values to strings according to a format.
- class GazeValidationData

  The gaze validation data structure
- class GazeValidationPoint

  A validation point.
- class LiveGazePoint

  The live gaze point used for verification during the calibration process.
- class UserPositionData

  The user position to be rendered on the screen.

Enumerations

- enum GazeOutputValue {
  DataTimeStamp = 0, **DataTimeStampReceived**, **DataTimeStampRelative**, **TrialId**,
  Tag, **CombinedGazePoint2dCompensatedX**, **CombinedGazePoint2dCompensatedY**, **Combined↩GazePoint2dX**,
  CombinedGazePoint2dY, **CombinedGazePoint2dIsValid**, **CombinedGazePoint3dCompensatedX**,
  CombinedGazePoint3dCompensatedY,
  CombinedGazePoint3dCompensatedZ, **CombinedGazePoint3dX**, **CombinedGazePoint3dY**, **Combined↩GazePoint3dZ**,
  CombinedGazePoint3dIsValid, **CombinedGazeOrigin3dX**, **CombinedGazeOrigin3dY**, **Combined↩GazeOrigin3dZ**,
  CombinedGazeOrigin3dIsValid, **CombinedGazeDistance**, **CombinedPupilDiameter**, **CombinedPupil↩DiameterIsValid**,
  LeftGazePoint2dX, **LeftGazePoint2dY**, **LeftGazePoint2dIsValid**, **LeftGazePoint3dX**,
  LeftGazePoint3dY, **LeftGazePoint3dZ**, **LeftGazePoint3dIsValid**, **LeftGazeOrigin3dX**,
  LeftGazeOrigin3dY, **LeftGazeOrigin3dZ**, **LeftGazeOrigin3dIsValid**, **LeftGazeDistance**,
  LeftPupilDiameter, **LeftPupilDiameterIsValid**, **RightGazePoint2dX**, **RightGazePoint2dY**,
  RightGazePoint2dIsValid, **RightGazePoint3dX**, **RightGazePoint3dY**, **RightGazePoint3dZ**,
  RightGazePoint3dIsValid, **RightGazeOrigin3dX**, **RightGazeOrigin3dY**, **RightGazeOrigin3dZ**,
  RightGazeOrigin3dIsValid, **RightGazeDistance**, **RightPupilDiameter**, **RightPupilDiameterIsValid** }

enummerates output values produced by the eyetracker

- enum CalibrationOutputValue {
  Point2dX, **Point2dY**, **LeftGazePoint2dX**, **LeftGazePoint2dY**,
  LeftGazePoint2dIsValid, **LeftAccuarcy**, **RightGazePoint2dX**, **RightGazePoint2dY**,
  RightGazePoint2dIsValid, **RightAccuarcy** }

  enummerates output values produced by the eyetracker
- enum ValidationOutputValue {
  Point2dX, **Point2dY**, **LeftAccuracy**, **LeftPrecision**,
  LeftPrecisionRMS, **RightAccuracy**, **RightPrecision**, **RightPrecisionRMS** }

  enummerates output values produced by the eyetracker

7.13.1 Enumeration Type Documentation

7.13.1.1 CalibrationOutputValue

`enum GazeUtilityLibrary.DataStructs.CalibrationOutputValue [strong]`

enummerates output values produced by the eyetracker

7.13.1.2 GazeOutputValue

`enum GazeUtilityLibrary.DataStructs.GazeOutputValue [strong]`

enummerates output values produced by the eyetracker

7.13.1.3 ValidationOutputValue

`enum GazeUtilityLibrary.DataStructs.ValidationOutputValue [strong]`

enummerates output values produced by the eyetracker

7.14 GazeUtilityLibrary.Tracker Namespace Reference

Classes

- class BaseTracker

  The common interface for the Tobii eyetracker Engines Core and Pro
- class CalibrationOrigin

  Helper class to hold the approximated gaze origin during the data collection of a calibration point.
- class EyeTrackerPro

  Interface to the Tobii SDK Pro engine
- class MouseTracker

  This class is used to hook into the system mouse events and track the position

7.15 ShowMouse Namespace Reference

Classes

- class App

  Interaction logic for App.xaml

7.16 Tobii Namespace Reference

7.17 Tobii.Research Namespace Reference

7.18 Tobii.Research.Addons Namespace Reference

Classes

- class CalibrationValidationPoint

  Represents a collected point that goes into the calibration validation. It contains calculated values for accuracy and precision as well as the original gaze samples collected for the point.
- class CalibrationValidationResult

  Contains the result of the calibration validation.
- class ScreenBasedCalibrationValidation

  Provides methods and properties for managing calibration validation for screen based eye trackers.

7.19 Tobii.Research.Addons.Utility Namespace Reference

Classes

- class **Extensions**

  Extensions with some operations on Point3D and NormalizedPoint2D among other things.
- class **TimeKeeper**

7.20 TobiiCalibrate Namespace Reference

Classes

- class App

  Interaction logic for App.xaml

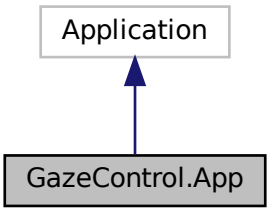
Chapter 8

Class Documentation

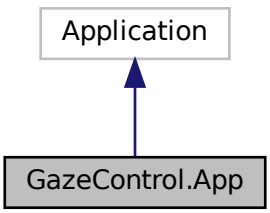
8.1 GazeControl.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for GazeControl.App:



Collaboration diagram for GazeControl.App:



8.1.1 Detailed Description

Interaction logic for App.xaml

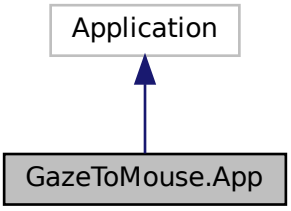
The documentation for this class was generated from the following file:

- source/GazeControl/App.xaml.cs

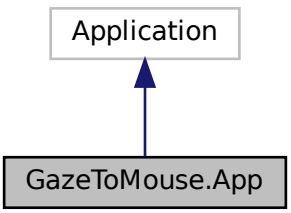
8.2 GazeToMouse.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for GazeToMouse.App:



Collaboration diagram for GazeToMouse.App:



Public Member Functions

- void GazeRecordingEnable ()

  Enable gaze recordings to disk.
- void GazeRecordingDisable ()

  Disable gaze recordings.
- void MouseTrackingEnable ()

  Enable mouse tracking which updates the mouse position to the current gaze point.
- void MouseTrackingDisable ()

  Disable mouse tracking.
- void ResetDriftCompensation ()

  Reset the current drift compensation offset to zero.
- double GetDriftDeviationAngle ()

  Get the drift deviation angle of the tracker.
- async Task< bool > CompensateDrift ()

  Start the drift compensation process
- async Task< bool > Loading ()

  Loading spinner.
- async Task< bool > CustomCalibrate ()

  Start the gaze calibration process
- async Task< bool > CalibrationValidate ()

  Start the gaze calibration process
- App ()

  Constructor: initialised logger, gaze configuration, pipe server, and calibration model

Properties

- TimeSpan StartTime `[get, set]`

  The start time of the application.
- TrackerLogger Logger `[get]`

  The logger handler.
- Dispatcher CustomDispatcher `[get]`

  The dispatcher to handle pipe commands.
- string LastTag `[get, set]`

  The last tag to annotate gaze data.
- string Tag `[get, set]`

  An arbitary tag to annotate gaze data.
- int TrialId `[get, set]`

  The trial ID to annotate gaze data.

8.2.1 Detailed Description

Interaction logic for App.xaml

8.2.2 Constructor & Destructor Documentation

8.2.2.1 App()

```
GazeToMouse.App.App ( )  [inline]
```

Constructor: initialised logger, gaze configuration, pipe server, and calibration model

8.2.3 Member Function Documentation

8.2.3.1 CalibrationValidate()

```
async Task<bool> GazeToMouse.App.CalibrationValidate ( )  [inline]
```

Start the gaze calibration process

Returns

True on success, false on failure

8.2.3.2 CompensateDrift()

```
async Task<bool> GazeToMouse.App.CompensateDrift ( )  [inline]
```

Start the drift compensation process

Returns

True on success, false on failure

8.2.3.3 CustomCalibrate()

```
async Task<bool> GazeToMouse.App.CustomCalibrate ( )  [inline]
```

Start the gaze calibration process

Returns

True on success, false on failure

8.2.3.4 GazeRecordingDisable()

```
void GazeToMouse.App.GazeRecordingDisable ( )  [inline]
```

Disable gaze recordings.

8.2.3.5 GazeRecordingEnable()

```
void GazeToMouse.App.GazeRecordingEnable ( )  [inline]
```

Enable gaze recordings to disk.

8.2.3.6 GetDriftDeviationAngle()

```
double GazeToMouse.App.GetDriftDeviationAngle ( )  [inline]
```

Get the drift deviation angle of the tracker.

Returns

The drift deviation angle.

8.2.3.7 Loading()

```
async Task<bool> GazeToMouse.App.Loading ( )  [inline]
```

Loading spinner.

Returns

True on success, false on failure

8.2.3.8 MouseTrackingDisable()

```
void GazeToMouse.App.MouseTrackingDisable ( )  [inline]
```

Disable mouse tracking.

8.2.3.9 MouseTrackingEnable()

```
void GazeToMouse.App.MouseTrackingEnable ( )  [inline]
```

Enable mouse tracking which updates the mouse position to the current gaze point.

8.2.3.10 ResetDriftCompensation()

```
void GazeToMouse.App.ResetDriftCompensation ( )  [inline]
```

Reset the current drift compensation offset to zero.

8.2.4 Property Documentation

8.2.4.1 CustomDispatcher

```
Dispatcher GazeToMouse.App.CustomDispatcher  [get]
```

The dispatcher to handle pipe commands.

8.2.4.2 LastTag

```
string GazeToMouse.App.LastTag  [get], [set]
```

The last tag to annotate gaze data.

8.2.4.3 Logger

```
TrackerLogger GazeToMouse.App.Logger  [get]
```

The logger handler.

8.2.4.4 StartTime

```
TimeSpan GazeToMouse.App.StartTime  [get], [set]
```

The start time of the application.

8.2.4.5 Tag

```
string GazeToMouse.App.Tag  [get], [set]
```

An arbitary tag to annotate gaze data.

8.2.4.6 TrialId

```
int GazeToMouse.App.TrialId  [get], [set]
```

The trial ID to annotate gaze data.

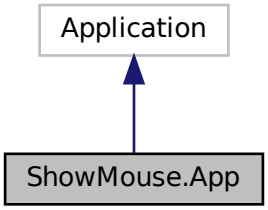
The documentation for this class was generated from the following file:

- source/GazeToMouse/App.xaml.cs

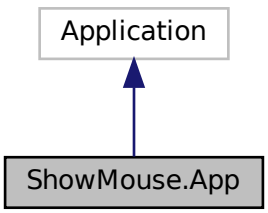
8.3 ShowMouse.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for ShowMouse.App:



Collaboration diagram for ShowMouse.App:



8.3.1 Detailed Description

Interaction logic for App.xaml

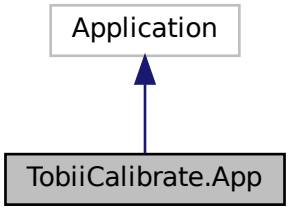
The documentation for this class was generated from the following file:

- source/ShowMouse/App.xaml.cs

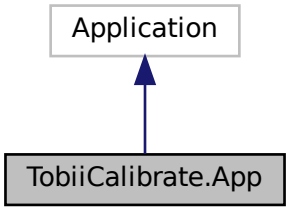
8.4 TobiiCalibrate.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for TobiiCalibrate.App:



Collaboration diagram for TobiiCalibrate.App:



8.4.1 Detailed Description

Interaction logic for App.xaml

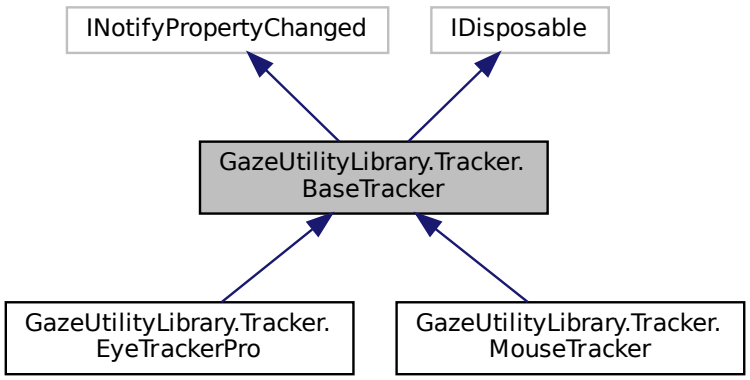
The documentation for this class was generated from the following file:

- source/TobiiCalibrate/App.xaml.cs

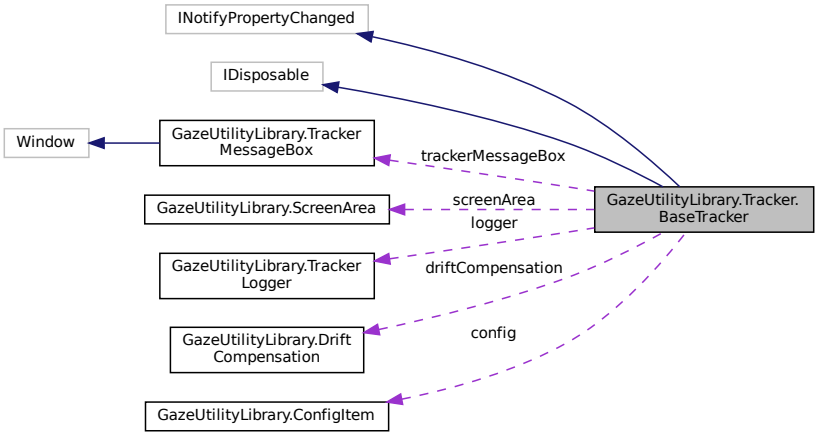
8.5 GazeUtilityLibrary.Tracker.BaseTracker Class Reference

The common interface for the Tobii eyetracker Engines Core and Pro

Inheritance diagram for GazeUtilityLibrary.Tracker.BaseTracker:



Collaboration diagram for GazeUtilityLibrary.Tracker.BaseTracker:



Public Types

- enum DeviceStatus {
  Configuring, **Initializing**, **InvalidConfiguration**, **DeviceNotConnected**,
  Tracking }

  The tracker device status

Public Member Functions

- delegate void GazeDataHandler (object sender, GazeData gazeData)

  Event handler for gaze data events of the eyetracker
- delegate void DriftCompensationEventHandler (object sender, Quaternion driftCompensation)

  Event handler for drift compensation events
- delegate void UserPositionDataHandler (object sender, UserPositionData e)

  Event handler for user position data events of the eyetracker
- BaseTracker (TrackerLogger logger, ConfigItem config, string deviceName)

  Initializes a new instance of the EyeTrackerHandler class.
- void Dispose ()

  Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
- virtual string PatternReplace (string pattern)

  Replaces a patten string with information from the eye tracker. This is device specific and may be overwritten by the device class.
- abstract Task InitCalibrationAsync ()

  Initialise the async calibartion process. This is device specific and must be overwritten by the device class.
- abstract void InitCalibration ()

  Initialise the calibartion process. This is device specific and must be overwritten by the device class.
- abstract void InitValidation ()

  Initialise the validation process. This is device specific and must be overwritten by the device class.
- abstract Task FinishCalibrationAsync ()

  Finish the async calibartion process. This is device specific and must be overwritten by the device class.
- abstract void FinishCalibration ()

  Finish the calibartion process. This is device specific and must be overwritten by the device class.
- abstract void FinishValidation ()

  Finish the validation process. This is device specific and must be overwritten by the device class.
- abstract Task< List< GazeCalibrationData > > ApplyCalibration ()

  Apply the calibration data. This is device specific and must be overwritten by the device class.
- abstract ? GazeValidationData ComputeValidation ()

  Apply the validation data. This is device specific and must be overwritten by the device class.
- abstract Task< bool > CollectCalibrationDataAsync (Point point)

  Collect calibration data on a calibration point. This is device specific and must be overwritten by the device class.
- abstract Task< bool > CollectValidationDataAsync (Point point)

  Collect validation data on a validation point. This is device specific and must be overwritten by the device class.
- void StartDriftCompensation ()

  Start the drift compensation process.
- void ResetDriftCompensation ()

  Reset the drift compensation value
- virtual bool IsInitialised ()

  Checks wheter the device is connected and initialised. This is device specific and may be overwritten. Otherwise true is always returned.

Public Attributes

- readonly string DeviceName

  The name of the tracker device

Protected Member Functions

- abstract void InitDriftCompensation ()

  Initialise the drift compensation. This is device specific and must be overwritten by the device class.
- abstract int GetFixationFrameCount (int durationThreshold)

  Get the number of required gaze samples to compute a fixation. This is device specific and must be overwritten by the device because the duration of fixation point detection depends on the frame rate of the device.
- abstract Vector3 GetUnitDirection ()

  Get the unit vector pointing in the direction of the gaze vector. This is device specific as the gaze data are represented in a coordinate system as defined by the device.
- virtual void Dispose (bool disposing)

  Releases unmanaged and - optionally - managed resources.
- bool IsReady ()

  Determines whether this eye tracker is ready.
- virtual void OnGazeDataReceived (GazeData gazeData)

  Called when [gaze data received].
- virtual void OnUserPositionDataReceived (UserPositionData e)

  Called when [user position data received].
- virtual void OnPropertyChanged (string property_name)

  Called when when the state property of EyeTracker is changing.
- virtual void OnTrackerDisabled (EventArgs e)

  Raises the E:TrackerDisabled event.
- void OnTrackerDisabledTimeout (object? source, ElapsedEventArgs e)

  Called after a specified amount of time of the eyetracker not being ready.
- virtual void OnTrackerEnabled (EventArgs e)

  Raises the E:TrackerEnabled event.

Protected Attributes

- Timer? dialogBoxTimer

  Timer to control the apperance of the dialog box
- TrackerLogger logger

  The logger
- TrackerMessageBox? trackerMessageBox

  The dialog box that is controlled by the dialogBoxTimer
- DriftCompensation? driftCompensation

  drift compensation handler
- ScreenArea? screenArea = null

  The scrren area structure holding the metrics of the screen in 3d space.
- ConfigItem config

  The gaze configuration item

Properties

- double??? DriftDeviationAngle `[get]`

  The deviation angle of the drift compensation.
- ScreenArea? ScreenArea `[get]`

  The scrren area structure holding the metrics of the screen in 3d space.
- DeviceStatus State `[get, set]`

  Gets or sets the state of the eyetracker device.

Events

- EventHandler? TrackerEnabled

  Occurs when [tracker enabled].
- EventHandler? TrackerDisabled

  Occurs when [tracker disabled].
- PropertyChangedEventHandler? PropertyChanged

  Occurs when a property value changes.
- GazeDataHandler? GazeDataReceived

  Occurs when [gaze data received].
- DriftCompensationEventHandler? DriftCompensationComputed

  Occurs when drift compensation was computed.
- UserPositionDataHandler? UserPositionDataReceived

  Occurs when [user position data received].

8.5.1 Detailed Description

The common interface for the Tobii eyetracker Engines Core and Pro

See also

INotifyPropertyChanged, IDisposable

8.5.2 Member Enumeration Documentation

8.5.2.1 DeviceStatus

```
enum GazeUtilityLibrary.Tracker.BaseTracker.DeviceStatus  [strong]
```

The tracker device status

8.5.3 Constructor & Destructor Documentation

8.5.3.1 BaseTracker()

```
GazeUtilityLibrary.Tracker.BaseTracker.BaseTracker (
            TrackerLogger logger,
            ConfigItem config,
            string deviceName )  [inline]
```

Initializes a new instance of the EyeTrackerHandler class.

Parameters

logger	The logger.
config	The configuration object.
deviceName	Name of the device.

8.5.4 Member Function Documentation

8.5.4.1 ApplyCalibration()

```
abstract Task<List<GazeCalibrationData> > GazeUtilityLibrary.Tracker.BaseTracker.Apply↩
Calibration ( )  [pure virtual]
```

Apply the calibration data. This is device specific and must be overwritten by the device class.

Returns

The calibration data result wrapped by an async handler.

Implemented in GazeUtilityLibrary.Tracker.EyeTrackerPro, and GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.2 CollectCalibrationDataAsync()

```
abstract Task<bool> GazeUtilityLibrary.Tracker.BaseTracker.CollectCalibrationDataAsync (
            Point point )  [pure virtual]
```

Collect calibration data on a calibration point. This is device specific and must be overwritten by the device class.

Parameters

point	The calibration point for which to collect data

Returns

True on success, false on failure, wrapped by an async handler.

Implemented in GazeUtilityLibrary.Tracker.MouseTracker, and GazeUtilityLibrary.Tracker.EyeTrackerPro.

8.5.4.3 CollectValidationDataAsync()

```
abstract Task<bool> GazeUtilityLibrary.Tracker.BaseTracker.CollectValidationDataAsync (
            Point point )  [pure virtual]
```

Collect validation data on a validation point. This is device specific and must be overwritten by the device class.

Parameters

point	The calibration point for which to collect data

Returns

True on success, false on failure, wrapped by an async handler.

Implemented in GazeUtilityLibrary.Tracker.EyeTrackerPro, and GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.4 ComputeValidation()

```
abstract ?  GazeValidationData GazeUtilityLibrary.Tracker.BaseTracker.ComputeValidation ( )
[pure virtual]
```

Apply the validation data. This is device specific and must be overwritten by the device class.

Returns

The validation data result.

Implemented in GazeUtilityLibrary.Tracker.EyeTrackerPro, and GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.5 Dispose() [1/2]

```
void GazeUtilityLibrary.Tracker.BaseTracker.Dispose ( )  [inline]
```

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

8.5.4.6 Dispose() [2/2]

```
virtual void GazeUtilityLibrary.Tracker.BaseTracker.Dispose (
            bool disposing )  [inline], [protected], [virtual]
```

Releases unmanaged and - optionally - managed resources.

Parameters

disposing	`true` to release both managed and unmanaged resources; `false` to release only unmanaged resources.

Reimplemented in GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.7 DriftCompensationEventHandler()

```
delegate void GazeUtilityLibrary.Tracker.BaseTracker.DriftCompensationEventHandler (
            object sender,
            Quaternion driftCompensation )
```

Event handler for drift compensation events

Parameters

sender	The sender.
driftCompensation	The drift compensation quaternion

8.5.4.8 FinishCalibration()

```
abstract void GazeUtilityLibrary.Tracker.BaseTracker.FinishCalibration ( )  [pure virtual]
```

Finish the calibartion process. This is device specific and must be overwritten by the device class.

Implemented in GazeUtilityLibrary.Tracker.EyeTrackerPro, and GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.9 FinishCalibrationAsync()

```
abstract Task GazeUtilityLibrary.Tracker.BaseTracker.FinishCalibrationAsync ( )  [pure virtual]
```

Finish the async calibartion process. This is device specific and must be overwritten by the device class.

Returns

An async handler

Implemented in GazeUtilityLibrary.Tracker.EyeTrackerPro, and GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.10 FinishValidation()

```
abstract void GazeUtilityLibrary.Tracker.BaseTracker.FinishValidation ( )  [pure virtual]
```

Finish the validation process. This is device specific and must be overwritten by the device class.

Implemented in GazeUtilityLibrary.Tracker.EyeTrackerPro, and GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.11 GazeDataHandler()

```
delegate void GazeUtilityLibrary.Tracker.BaseTracker.GazeDataHandler (
            object sender,
            GazeData gazeData )
```

Event handler for gaze data events of the eyetracker

Parameters

sender	The sender.
gazeData	The e.

8.5.4.12 GetFixationFrameCount()

```
abstract int GazeUtilityLibrary.Tracker.BaseTracker.GetFixationFrameCount (
            int durationThreshold )  [protected], [pure virtual]
```

Get the number of required gaze samples to compute a fixation. This is device specific and must be overwritten by the device because the duration of fixation point detection depends on the frame rate of the device.

Parameters

durationThreshold	The required fixation duration in milliseconds.

Returns

The number of gaze samples to require for fixation detection.

Implemented in GazeUtilityLibrary.Tracker.EyeTrackerPro, and GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.13 GetUnitDirection()

```
abstract Vector3 GazeUtilityLibrary.Tracker.BaseTracker.GetUnitDirection ( )  [protected],
[pure virtual]
```

Get the unit vector pointing in the direction of the gaze vector. This is device specific as the gaze data are represented in a coordinate system as defined by the device.

Returns

The unit vector

Implemented in GazeUtilityLibrary.Tracker.EyeTrackerPro, and GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.14 InitCalibration()

```
abstract void GazeUtilityLibrary.Tracker.BaseTracker.InitCalibration ( )  [pure virtual]
```

Initialise the calibartion process. This is device specific and must be overwritten by the device class.

Implemented in GazeUtilityLibrary.Tracker.MouseTracker, and GazeUtilityLibrary.Tracker.EyeTrackerPro.

8.5.4.15 InitCalibrationAsync()

```
abstract Task GazeUtilityLibrary.Tracker.BaseTracker.InitCalibrationAsync ( )  [pure virtual]
```

Initialise the async calibartion process. This is device specific and must be overwritten by the device class.

Returns

An async handler

Implemented in GazeUtilityLibrary.Tracker.MouseTracker, and GazeUtilityLibrary.Tracker.EyeTrackerPro.

8.5.4.16 InitDriftCompensation()

```
abstract void GazeUtilityLibrary.Tracker.BaseTracker.InitDriftCompensation ( )  [protected],
[pure virtual]
```

Initialise the drift compensation. This is device specific and must be overwritten by the device class.

Implemented in GazeUtilityLibrary.Tracker.EyeTrackerPro, and GazeUtilityLibrary.Tracker.MouseTracker.

8.5.4.17 InitValidation()

```
abstract void GazeUtilityLibrary.Tracker.BaseTracker.InitValidation ( )  [pure virtual]
```

Initialise the validation process. This is device specific and must be overwritten by the device class.

Implemented in GazeUtilityLibrary.Tracker.MouseTracker, and GazeUtilityLibrary.Tracker.EyeTrackerPro.

8.5.4.18 IsInitialised()

```
virtual bool GazeUtilityLibrary.Tracker.BaseTracker.IsInitialised ( )  [inline], [virtual]
```

Checks wheter the device is connected and initialised. This is device specific and may be overwritten. Otherwise true is always returned.

Returns

True

Reimplemented in GazeUtilityLibrary.Tracker.EyeTrackerPro.

8.5.4.19 IsReady()

```
bool GazeUtilityLibrary.Tracker.BaseTracker.IsReady ( )  [inline], [protected]
```

Determines whether this eye tracker is ready.

Returns

`true` if this instance is ready; otherwise, `false`.

8.5.4.20 OnGazeDataReceived()

```
virtual void GazeUtilityLibrary.Tracker.BaseTracker.OnGazeDataReceived (
            GazeData gazeData )  [inline], [protected], [virtual]
```

Called when [gaze data received].

Parameters

gazeData	The gaze data event data.

8.5.4.21 OnPropertyChanged()

```
virtual void GazeUtilityLibrary.Tracker.BaseTracker.OnPropertyChanged (
            string property_name )  [inline], [protected], [virtual]
```

Called when when the state property of EyeTracker is changing.

Parameters

property_name	Name of the property in WPF.

8.5.4.22 OnTrackerDisabled()

```
virtual void GazeUtilityLibrary.Tracker.BaseTracker.OnTrackerDisabled (
            EventArgs e )  [inline], [protected], [virtual]
```

Raises the E:TrackerDisabled event.

Parameters

e	The EventArgs instance containing the event data.

8.5.4.23 OnTrackerDisabledTimeout()

```
void GazeUtilityLibrary.Tracker.BaseTracker.OnTrackerDisabledTimeout (
            object?  source,
            ElapsedEventArgs e )  [inline], [protected]
```

Called after a specified amount of time of the eyetracker not being ready.

Parameters

source	The source.
e	The ElapsedEventArgs instance containing the event data.

8.5.4.24 OnTrackerEnabled()

```
virtual void GazeUtilityLibrary.Tracker.BaseTracker.OnTrackerEnabled (
            EventArgs e )  [inline], [protected], [virtual]
```

Raises the E:TrackerEnabled event.

Parameters

e	The EventArgs instance containing the event data.

8.5.4.25 OnUserPositionDataReceived()

```
virtual void GazeUtilityLibrary.Tracker.BaseTracker.OnUserPositionDataReceived (
            UserPositionData e )  [inline], [protected], [virtual]
```

Called when [user position data received].

Parameters

e	The gaze data event data.

8.5.4.26 PatternReplace()

```
virtual string GazeUtilityLibrary.Tracker.BaseTracker.PatternReplace (
            string pattern )  [inline], [virtual]
```

Replaces a patten string with information from the eye tracker. This is device specific and may be overwritten by the device class.

Returns

The string where patterns were replaced.

Reimplemented in GazeUtilityLibrary.Tracker.EyeTrackerPro.

8.5.4.27 ResetDriftCompensation()

```
void GazeUtilityLibrary.Tracker.BaseTracker.ResetDriftCompensation ( )  [inline]
```

Reset the drift compensation value

8.5.4.28 StartDriftCompensation()

```
void GazeUtilityLibrary.Tracker.BaseTracker.StartDriftCompensation ( )  [inline]
```

Start the drift compensation process.

8.5.4.29 UserPositionDataHandler()

```
delegate void GazeUtilityLibrary.Tracker.BaseTracker.UserPositionDataHandler (
            object sender,
            UserPositionData e )
```

Event handler for user position data events of the eyetracker

Parameters

sender	The sender.
e	The e.

8.5.5 Member Data Documentation

8.5.5.1 config

```
ConfigItem GazeUtilityLibrary.Tracker.BaseTracker.config  [protected]
```

The gaze configuration item

8.5.5.2 DeviceName

```
readonly string GazeUtilityLibrary.Tracker.BaseTracker.DeviceName
```

The name of the tracker device

8.5.5.3 dialogBoxTimer

```
Timer?  GazeUtilityLibrary.Tracker.BaseTracker.dialogBoxTimer  [protected]
```

Timer to control the apperance of the dialog box

8.5.5.4 driftCompensation

```
DriftCompensation?  GazeUtilityLibrary.Tracker.BaseTracker.driftCompensation  [protected]
```

drift compensation handler

8.5.5.5 logger

```
TrackerLogger GazeUtilityLibrary.Tracker.BaseTracker.logger  [protected]
```

The logger

8.5.5.6 screenArea

```
ScreenArea?  GazeUtilityLibrary.Tracker.BaseTracker.screenArea = null  [protected]
```

The scrren area structure holding the metrics of the screen in 3d space.

8.5.5.7 trackerMessageBox

```
TrackerMessageBox?  GazeUtilityLibrary.Tracker.BaseTracker.trackerMessageBox  [protected]
```

The dialog box that is controlled by the dialogBoxTimer

8.5.6 Property Documentation

8.5.6.1 DriftDeviationAngle

```
double??? GazeUtilityLibrary.Tracker.BaseTracker.DriftDeviationAngle [get]
```

The deviation angle of the drift compensation.

8.5.6.2 ScreenArea

```
ScreenArea? GazeUtilityLibrary.Tracker.BaseTracker.ScreenArea [get]
```

The scrren area structure holding the metrics of the screen in 3d space.

8.5.6.3 State

```
DeviceStatus GazeUtilityLibrary.Tracker.BaseTracker.State [get], [set]
```

Gets or sets the state of the eyetracker device.

The state.

8.5.7 Event Documentation

8.5.7.1 DriftCompensationComputed

```
DriftCompensationEventHandler? GazeUtilityLibrary.Tracker.BaseTracker.DriftCompensation↩
Computed
```

Occurs when drift compensation was computed.

8.5.7.2 GazeDataReceived

```
GazeDataHandler? GazeUtilityLibrary.Tracker.BaseTracker.GazeDataReceived
```

Occurs when [gaze data received].

8.5.7.3 PropertyChanged

```
PropertyChangedEventHandler?  GazeUtilityLibrary.Tracker.BaseTracker.PropertyChanged
```

Occurs when a property value changes.

8.5.7.4 TrackerDisabled

```
EventHandler?  GazeUtilityLibrary.Tracker.BaseTracker.TrackerDisabled
```

Occurs when [tracker disabled].

8.5.7.5 TrackerEnabled

```
EventHandler?  GazeUtilityLibrary.Tracker.BaseTracker.TrackerEnabled
```

Occurs when [tracker enabled].

8.5.7.6 UserPositionDataReceived

```
UserPositionDataHandler?  GazeUtilityLibrary.Tracker.BaseTracker.UserPositionDataReceived
```

Occurs when [user position data received].

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Tracker/BaseTracker.cs

8.6 CustomCalibrationLibrary.Extensions.BrushExtension Class Reference

Allows to attach a brush property.

Static Public Member Functions

- static Brush GetBrush (DependencyObject target)

  Get the brush property value of the target.
- static void SetBrush (DependencyObject target, Brush value)

  Setting the brush property value of a target

Static Public Attributes

- static DependencyProperty BrushProperty = DependencyProperty.RegisterAttached("Brush", typeof(Brush), typeof(BrushExtension), new PropertyMetadata(null))

  The brush color dependency property.

8.6.1 Detailed Description

Allows to attach a brush property.

8.6.2 Member Function Documentation

8.6.2.1 GetBrush()

```
static Brush CustomCalibrationLibrary.Extensions.BrushExtension.GetBrush (
            DependencyObject target )  [inline], [static]
```

Get the brush property value of the target.

Parameters

target	The target.

Returns

The brush of the target.

8.6.2.2 SetBrush()

```
static void CustomCalibrationLibrary.Extensions.BrushExtension.SetBrush (
            DependencyObject target,
            Brush value )  [inline], [static]
```

Setting the brush property value of a target

Parameters

target	The target.
value	The brush property value.

8.6.3 Member Data Documentation

8.6.3.1 BrushProperty

```
DependencyProperty CustomCalibrationLibrary.Extensions.BrushExtension.BrushProperty = Dependency↩
Property.RegisterAttached("Brush", typeof(Brush), typeof(BrushExtension), new PropertyMetadata(null))
[static]
```

The brush color dependency property.

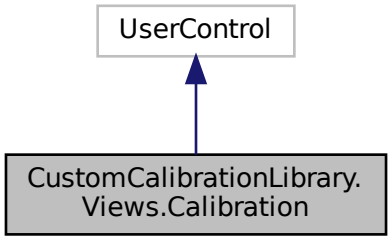
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Extensions/BrushExtension.cs

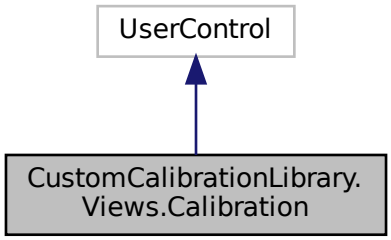
8.7 CustomCalibrationLibrary.Views.Calibration Class Reference

Interaction logic for Calibration.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.Calibration:



Collaboration diagram for CustomCalibrationLibrary.Views.Calibration:



Public Member Functions

- Calibration (CalibrationModel model)

  Initializes a new instance of the Calibration class.

8.7.1 Detailed Description

Interaction logic for Calibration.xaml

8.7.2 Constructor & Destructor Documentation

8.7.2.1 Calibration()

```
CustomCalibrationLibrary.Views.Calibration.Calibration (
            CalibrationModel model )  [inline]
```

Initializes a new instance of the Calibration class.

Parameters

model	The calibration model

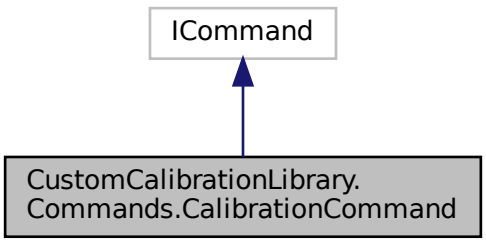
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/Calibration.xaml.cs

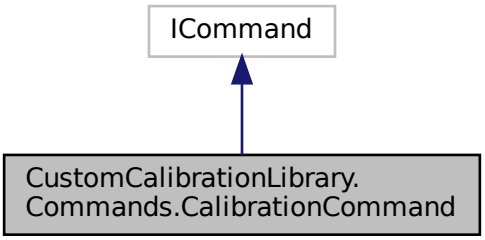
8.8 CustomCalibrationLibrary.Commands.CalibrationCommand Class Reference

Comand class to trigger calibration events.

Inheritance diagram for CustomCalibrationLibrary.Commands.CalibrationCommand:



Collaboration diagram for CustomCalibrationLibrary.Commands.CalibrationCommand:



Public Member Functions

- CalibrationCommand (CalibrationModel model, CalibrationEventType eventType)

  Initializes a new instance of the CalibrationCommand class.
- bool CanExecute (object? parameter)

  Returns whether command can be executed or not.
- void Execute (object? parameter)

  Send calibration event.

Properties

- EventHandler? CanExecuteChanged

  Event handler on can executed flag change.

8.8.1 Detailed Description

Comand class to trigger calibration events.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 CalibrationCommand()

```
CustomCalibrationLibrary.Commands.CalibrationCommand.CalibrationCommand (
            CalibrationModel model,
            CalibrationEventType eventType )  [inline]
```

Initializes a new instance of the CalibrationCommand class.

Parameters

model	The calibration model
eventType	The type of the calibration event.

8.8.3 Member Function Documentation

8.8.3.1 CanExecute()

```
bool CustomCalibrationLibrary.Commands.CalibrationCommand.CanExecute (
            object?  parameter )  [inline]
```

Returns whether command can be executed or not.

Parameters

parameter	The command parameter

Returns

True

8.8.3.2 Execute()

```
void CustomCalibrationLibrary.Commands.CalibrationCommand.Execute (
            object?  parameter )  [inline]
```

Send calibration event.

Parameters

parameter	The command parameter

8.8.4 Property Documentation

8.8.4.1 CanExecuteChanged

```
EventHandler?  CustomCalibrationLibrary.Commands.CalibrationCommand.CanExecuteChanged  [add],
[remove]
```

Event handler on can executed flag change.

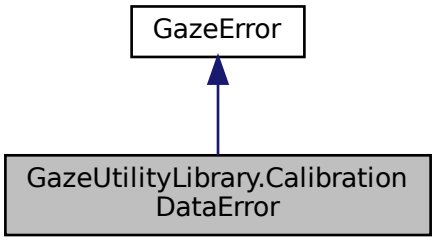
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Commands/CalibrationCommand.cs

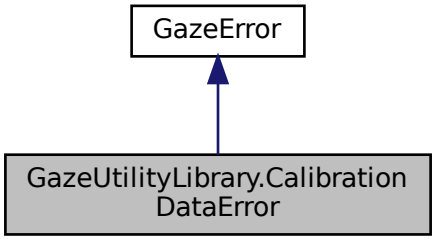
8.9 GazeUtilityLibrary.CalibrationDataError Class Reference

The calibration data error class to convert error flags to binary strings.

Inheritance diagram for GazeUtilityLibrary.CalibrationDataError:



Collaboration diagram for GazeUtilityLibrary.CalibrationDataError:



Public Member Functions

- string GetCalibrationDataErrorString ()

  Gets the gaze error string.

Properties

- ECalibrationDataError Error [set]

  The error flags.

Additional Inherited Members

8.9.1 Detailed Description

The calibration data error class to convert error flags to binary strings.

8.9.2 Member Function Documentation

8.9.2.1 GetCalibrationDataErrorString()

```
string GazeUtilityLibrary.CalibrationDataError.GetCalibrationDataErrorString ( )  [inline]
```

Gets the gaze error string.

Returns

the error string with binary error values if errors ocurred, the empty srting otherwise

8.9.3 Property Documentation

8.9.3.1 Error

```
ECalibrationDataError GazeUtilityLibrary.CalibrationDataError.Error  [set]
```

The error flags.

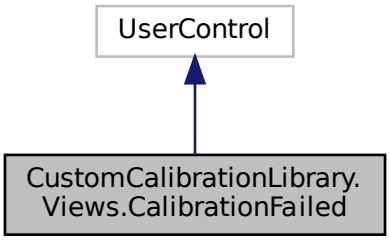
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeError.cs

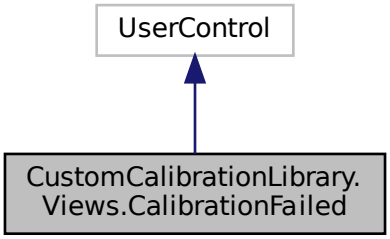
8.10 CustomCalibrationLibrary.Views.CalibrationFailed Class Reference

Interaction logic for CalibrationFailed.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationFailed:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationFailed:



Public Member Functions

- CalibrationFailed (CalibrationModel model)
  Constructor

8.10.1 Detailed Description

Interaction logic for CalibrationFailed.xaml

8.10.2 Constructor & Destructor Documentation

8.10.2.1 CalibrationFailed()

```
CustomCalibrationLibrary.Views.CalibrationFailed.CalibrationFailed (
            CalibrationModel model ) [inline]
```

Constructor

Parameters

model	The claibration model

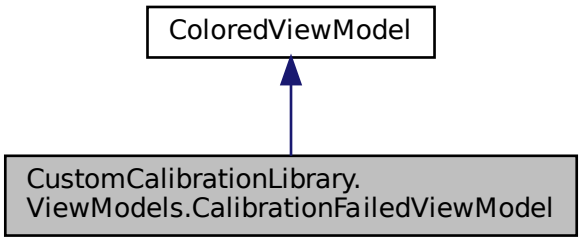
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/CalibrationFailed.xaml.cs

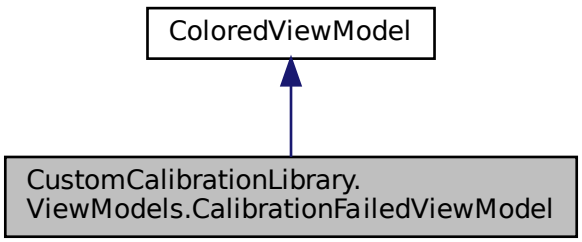
8.11 CustomCalibrationLibrary.ViewModels.CalibrationFailedViewModel Class Reference

The view model class for a failed calibration.

Inheritance diagram for CustomCalibrationLibrary.ViewModels.CalibrationFailedViewModel:



Collaboration diagram for CustomCalibrationLibrary.ViewModels.CalibrationFailedViewModel:



Public Member Functions

- CalibrationFailedViewModel (CalibrationModel model)

  Constructor

Properties

- ICommand CalibrationRestartCommand `[get]`

  Command to restart the calibration
- ICommand CalibrationAbortCommand `[get]`

  Command to abort the calibration
- string Error `[get, set]`

  The error message to be updated on the view.

Events

- PropertyChangedEventHandler? PropertyChanged

  The property change event to update the view.

8.11.1 Detailed Description

The view model class for a failed calibration.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 CalibrationFailedViewModel()

```
CustomCalibrationLibrary.ViewModels.CalibrationFailedViewModel.CalibrationFailedViewModel (
            CalibrationModel model ) [inline]
```

Constructor

Parameters

model	The calibration model

8.11.3 Property Documentation

8.11.3.1 CalibrationAbortCommand

```
ICommand CustomCalibrationLibrary.ViewModels.CalibrationFailedViewModel.CalibrationAbort↩
Command  [get]
```

Command to abort the calibration

8.11.3.2 CalibrationRestartCommand

```
ICommand CustomCalibrationLibrary.ViewModels.CalibrationFailedViewModel.CalibrationRestart↩
Command  [get]
```

Command to restart the calibration

8.11.3.3 Error

```
string CustomCalibrationLibrary.ViewModels.CalibrationFailedViewModel.Error  [get], [set]
```

The error message to be updated on the view.

8.11.4 Event Documentation

8.11.4.1 PropertyChanged

```
PropertyChangedEventHandler?  CustomCalibrationLibrary.ViewModels.CalibrationFailedView↩
Model.PropertyChanged
```

The property change event to update the view.

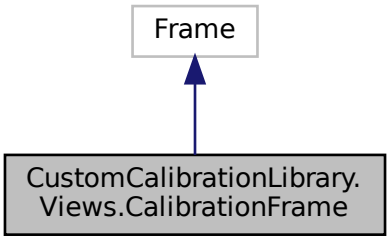
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/CalibrationFailedViewModel.cs

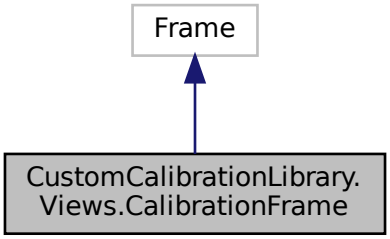
8.12 CustomCalibrationLibrary.Views.CalibrationFrame Class Reference

Interaction logic for CalibrationCollection.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationFrame:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationFrame:



Public Member Functions

- CalibrationFrame (CalibrationModel model, Window window)

  Initializes a new instance of the CalibrationFrame class.

8.12.1 Detailed Description

Interaction logic for CalibrationCollection.xaml

8.12.2 Constructor & Destructor Documentation

8.12.2.1 CalibrationFrame()

```
CustomCalibrationLibrary.Views.CalibrationFrame.CalibrationFrame (
            CalibrationModel model,
            Window window )  [inline]
```

Initializes a new instance of the CalibrationFrame class.

Parameters

model	The calibration model.
window	The target window.

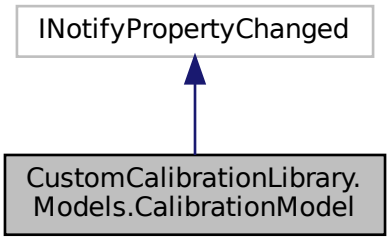
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/CalibrationFrame.xaml.cs

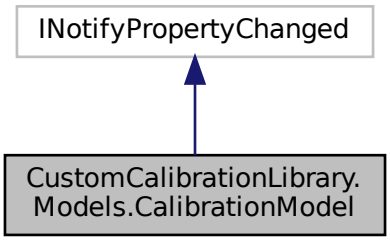
8.13 CustomCalibrationLibrary.Models.CalibrationModel Class Reference

The model for the calibration process.

Inheritance diagram for CustomCalibrationLibrary.Models.CalibrationModel:



Collaboration diagram for CustomCalibrationLibrary.Models.CalibrationModel:



Public Member Functions

- void OnCalibrationEvent (CalibrationEventType type)

  The calibraion event change handler.
- CalibrationModel (TrackerLogger logger, double[ ][ ] points, Color backgroundColor, Color frameColor, double accuracyThreshold, double precisionThreshold, int retries)

  Constructor
- void UpdateGazePoint (double x, double y)

  Update the normalized gaze point on the screen.
- void PrepareCalibration ()

  Prepare the calibration window.
- void NextCalibrationPoint ()

  Trigger the next calibration point.
- void RedoCalibrationPoint ()

  Remove and re-add the current calibration point
- void GazeDataCollected ()

  Trigger the data collected events.
- void GazeDataCollectionFailed ()

  Annotate calibration point as failed point.
- void SetCalibrationResult (List< GazeCalibrationData > points)

  Updates the calibration results on the screen.

Properties

- int Retries `[get]`

  The number of automatic calibration retries to perform.
- int RetryCount `[get, set]`

  The automatic aclibration retry counter.
- double AccuracyThreshold `[get]`

  The accuracy threshold.
- double PrecisionThreshold `[get]`

  The precision threshold.
- Color BackgroundColor `[get]`

The background color of the canvas.

- Color FrameColor [get]

  The background color of the frame.
- double CalibrationAccuracyLeft [get]

  The approximated accuracy of the current calibration for the left eye.
- double CalibrationAccuracyRight [get]

  The approximated accuracy of the current calibration for the right eye.
- string Error [get, set]

  The error message of the calibration process.
- GazeValidationData ValidationData [get, set]

  The data returned by a successful validation process.
- Cursor CursorType [get, set]

  The data returned by a successful validation process.
- CalibrationStatus Status [get, set]

  The status of the calibarion process.
- CalibrationStatus LastStatus [get]

  The calibration status before an error occured.
- Point[] Points [get]

  All calibration points.
- ObservableCollection< CalibrationPoint > CalibrationPoints [get]

  The calibration points to be added during the calibration process.
- Point GazePoint [get]

  The gaze point position.
- UserPositionData UserPositionGuide [get, set]

  The user position giude values.
- int Index [get]

  The index of the current calibration point

Events

- EventHandler< CalibrationEventType >? CalibrationEvent

  Event to trigger changes in the calibration process.
- PropertyChangedEventHandler? PropertyChanged

  Event to trigger property changes in this class.
- EventHandler< Point >? GazePointChanged

  Event to trigger gaze point changes.
- EventHandler< UserPositionData >? UserPositionGuideChanged

  Event to trigger user position guide changes.

8.13.1 Detailed Description

The model for the calibration process.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 CalibrationModel()

```
CustomCalibrationLibrary.Models.CalibrationModel.CalibrationModel (
            TrackerLogger logger,
            double points[][],
            Color backgroundColor,
            Color frameColor,
            double accuracyThreshold,
            double precisionThreshold,
            int retries )  [inline]
```

Constructor

Parameters

logger	The log handler
points	Calibration points
backgroundColor	The background color of the canvas
frameColor	The background color if the user interaction frame
accuracyThreshold	The accuracy threshold
precisionThreshold	The precision threshold
retries	The number of automatic retries

8.13.3 Member Function Documentation

8.13.3.1 GazeDataCollected()

```
void CustomCalibrationLibrary.Models.CalibrationModel.GazeDataCollected ( )  [inline]
```

Trigger the data collected events.

8.13.3.2 GazeDataCollectionFailed()

```
void CustomCalibrationLibrary.Models.CalibrationModel.GazeDataCollectionFailed ( )  [inline]
```

Annotate calibration point as failed point.

8.13.3.3 NextCalibrationPoint()

```
void CustomCalibrationLibrary.Models.CalibrationModel.NextCalibrationPoint ( )  [inline]
```

Trigger the next calibration point.

8.13.3.4 OnCalibrationEvent()

```
void CustomCalibrationLibrary.Models.CalibrationModel.OnCalibrationEvent (
            CalibrationEventType type )  [inline]
```

The calibraion event change handler.

Parameters

type	

8.13.3.5 PrepareCalibration()

```
void CustomCalibrationLibrary.Models.CalibrationModel.PrepareCalibration ( )  [inline]
```

Prepare the calibration window.

8.13.3.6 RedoCalibrationPoint()

```
void CustomCalibrationLibrary.Models.CalibrationModel.RedoCalibrationPoint ( )  [inline]
```

Remove and re-add the current calibration point

8.13.3.7 SetCalibrationResult()

```
void CustomCalibrationLibrary.Models.CalibrationModel.SetCalibrationResult (
            List< GazeCalibrationData > points )  [inline]
```

Updates the calibration results on the screen.

Parameters

points	

8.13.3.8 UpdateGazePoint()

```
void CustomCalibrationLibrary.Models.CalibrationModel.UpdateGazePoint (
            double x,
            double y )  [inline]
```

Update the normalized gaze point on the screen.

Parameters

x	The x coordinate
y	The y coordinate

8.13.4 Property Documentation

8.13.4.1 AccuracyThreshold

```
double CustomCalibrationLibrary.Models.CalibrationModel.AccuracyThreshold [get]
```

The accuracy threshold.

8.13.4.2 BackgroundColor

```
Color CustomCalibrationLibrary.Models.CalibrationModel.BackgroundColor [get]
```

The background color of the canvas.

8.13.4.3 CalibrationAccuracyLeft

```
double CustomCalibrationLibrary.Models.CalibrationModel.CalibrationAccuracyLeft [get]
```

The approximated accuracy of the current calibration for the left eye.

8.13.4.4 CalibrationAccuracyRight

```
double CustomCalibrationLibrary.Models.CalibrationModel.CalibrationAccuracyRight [get]
```

The approximated accuracy of the current calibration for the right eye.

8.13.4.5 CalibrationPoints

```
ObservableCollection<CalibrationPoint> CustomCalibrationLibrary.Models.CalibrationModel.↩
CalibrationPoints [get]
```

The calibration points to be added during the calibration process.

8.13.4.6 CursorType

```
Cursor CustomCalibrationLibrary.Models.CalibrationModel.CursorType  [get], [set]
```

The data returned by a successful validation process.

8.13.4.7 Error

```
string CustomCalibrationLibrary.Models.CalibrationModel.Error  [get], [set]
```

The error message of the calibration process.

8.13.4.8 FrameColor

```
Color CustomCalibrationLibrary.Models.CalibrationModel.FrameColor  [get]
```

The background color of the frame.

8.13.4.9 GazePoint

```
Point CustomCalibrationLibrary.Models.CalibrationModel.GazePoint  [get]
```

The gaze point position.

8.13.4.10 Index

```
int CustomCalibrationLibrary.Models.CalibrationModel.Index  [get]
```

The index of the current calibration point

8.13.4.11 LastStatus

```
CalibrationStatus CustomCalibrationLibrary.Models.CalibrationModel.LastStatus  [get]
```

The calibration status before an error occured.

8.13.4.12 Points

```
Point [] CustomCalibrationLibrary.Models.CalibrationModel.Points  [get]
```

All calibration points.

8.13.4.13 PrecisionThreshold

```
double CustomCalibrationLibrary.Models.CalibrationModel.PrecisionThreshold  [get]
```

The precision threshold.

8.13.4.14 Retries

```
int CustomCalibrationLibrary.Models.CalibrationModel.Retries  [get]
```

The number of automatic calibration retries to perform.

8.13.4.15 RetryCount

```
int CustomCalibrationLibrary.Models.CalibrationModel.RetryCount  [get], [set]
```

The automatic aclibration retry counter.

8.13.4.16 Status

```
CalibrationStatus CustomCalibrationLibrary.Models.CalibrationModel.Status  [get], [set]
```

The status of the calibarion process.

8.13.4.17 UserPositionGuide

```
UserPositionData CustomCalibrationLibrary.Models.CalibrationModel.UserPositionGuide  [get],
[set]
```

The user position giude values.

8.13.4.18 ValidationData

GazeValidationData CustomCalibrationLibrary.Models.CalibrationModel.ValidationData [get], [set]

The data returned by a successful validation process.

8.13.5 Event Documentation

8.13.5.1 CalibrationEvent

EventHandler<CalibrationEventType>? CustomCalibrationLibrary.Models.CalibrationModel.Calibration↩ Event

Event to trigger changes in the calibration process.

8.13.5.2 GazePointChanged

EventHandler<Point>? CustomCalibrationLibrary.Models.CalibrationModel.GazePointChanged

Event to trigger gaze point changes.

8.13.5.3 PropertyChanged

PropertyChangedEventHandler? CustomCalibrationLibrary.Models.CalibrationModel.PropertyChanged

Event to trigger property changes in this class.

8.13.5.4 UserPositionGuideChanged

EventHandler<UserPositionData>? CustomCalibrationLibrary.Models.CalibrationModel.UserPosition↩ GuideChanged

Event to trigger user position guide changes.

The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Models/CalibrationModel.cs

8.14 GazeUtilityLibrary.Tracker.CalibrationOrigin Class Reference

Helper class to hold the approximated gaze origin during the data collection of a calibration point.

Public Member Functions

- CalibrationOrigin (Point3D left, Point3D right, NormalizedPoint2D calibrationPoint)

  Constructor

Properties

- NormalizedPoint2D CalibrationPoint `[get]`

  The calibration point
- Point3D Left `[get]`

  The approximated gaze origin of the left eye.
- Point3D Right `[get]`

  The approximated gaze origin of the right eye.

8.14.1 Detailed Description

Helper class to hold the approximated gaze origin during the data collection of a calibration point.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 CalibrationOrigin()

```
GazeUtilityLibrary.Tracker.CalibrationOrigin.CalibrationOrigin (
            Point3D left,
            Point3D right,
            NormalizedPoint2D calibrationPoint )  [inline]
```

Constructor

Parameters

left	The approximated gaze origin of the left eye.
right	The approximated gaze origin of the right eye.
calibrationPoint	The calibration point

8.14.3 Property Documentation

8.14.3.1 CalibrationPoint

```
NormalizedPoint2D GazeUtilityLibrary.Tracker.CalibrationOrigin.CalibrationPoint  [get]
```

The calibration point

8.14.3.2 Left

```
Point3D GazeUtilityLibrary.Tracker.CalibrationOrigin.Left  [get]
```

The approximated gaze origin of the left eye.

8.14.3.3 Right

```
Point3D GazeUtilityLibrary.Tracker.CalibrationOrigin.Right  [get]
```

The approximated gaze origin of the right eye.

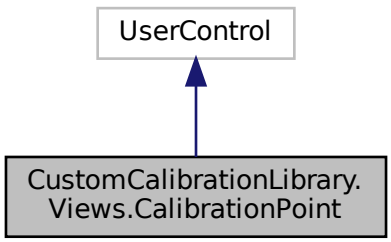
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Tracker/EyeTrackerPro.cs

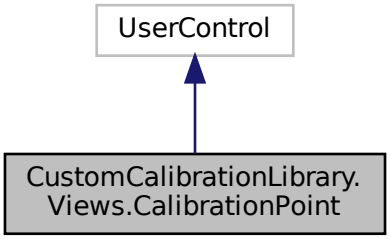
8.15 CustomCalibrationLibrary.Views.CalibrationPoint Class Reference

Interaction logic for CalibrationPoint.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationPoint:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationPoint:



Public Member Functions

- CalibrationPoint ()

  Initializes a new instance of the CalibrationPoint class.

8.15.1 Detailed Description

Interaction logic for CalibrationPoint.xaml

8.15.2 Constructor & Destructor Documentation

8.15.2.1 CalibrationPoint()

```
CustomCalibrationLibrary.Views.CalibrationPoint.CalibrationPoint ( )  [inline]
```

Initializes a new instance of the CalibrationPoint class.

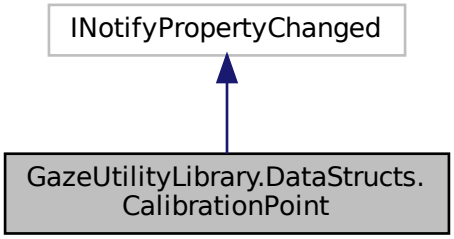
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/CalibrationPoint.xaml.cs

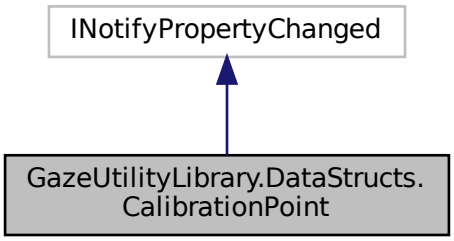
8.16 GazeUtilityLibrary.DataStructs.CalibrationPoint Class Reference

A calibration point class holding several metrics connected to a calibration point.

Inheritance diagram for GazeUtilityLibrary.DataStructs.CalibrationPoint:



Collaboration diagram for GazeUtilityLibrary.DataStructs.CalibrationPoint:



Public Member Functions

- CalibrationPoint (Point position, int index)

  Initializes a new instance of the CalibrationPoint class.

Properties

- int Index `[get]`

  The index of the calibration point.
- bool HasData `[get, set]`

  Flag to indicate whether data has been collected for this calibration point.
- bool HasFailed `[get, set]`

  Flag to indicate whether data has been collected for this calibration point.

- Point Position `[get, set]`

  The position of the calibration point.
- Point GazePositionAverage `[get, set]`

  The average between the left and the right gaze point.
- Point GazePositionAverageDelta `[get]`

  The delta from the calibration point of the average between the left and the right gaze point.
- Point GazePositionLeft `[get, set]`

  The left gaze point.
- Point GazePositionLeftDelta `[get]`

  The delta from the calibration point of the left gaze point.
- Point GazePositionRight `[get, set]`

  The right gaze point.
- Point GazePositionRightDelta `[get]`

  The delta from the calibration point of the right gaze point.

Events

- PropertyChangedEventHandler? PropertyChanged

  Event to trigger property changes.

8.16.1 Detailed Description

A calibration point class holding several metrics connected to a calibration point.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 CalibrationPoint()

```
GazeUtilityLibrary.DataStructs.CalibrationPoint.CalibrationPoint (
            Point position,
            int index )  [inline]
```

Initializes a new instance of the CalibrationPoint class.

Parameters

position	The position of the calibration point.
index	The index of the calibration point.

8.16.3 Property Documentation

8.16.3.1 GazePositionAverage

```
Point GazeUtilityLibrary.DataStructs.CalibrationPoint.GazePositionAverage [get], [set]
```

The average between the left and the right gaze point.

8.16.3.2 GazePositionAverageDelta

```
Point GazeUtilityLibrary.DataStructs.CalibrationPoint.GazePositionAverageDelta [get]
```

The delta from the calibration point of the average between the left and the right gaze point.

8.16.3.3 GazePositionLeft

```
Point GazeUtilityLibrary.DataStructs.CalibrationPoint.GazePositionLeft [get], [set]
```

The left gaze point.

8.16.3.4 GazePositionLeftDelta

```
Point GazeUtilityLibrary.DataStructs.CalibrationPoint.GazePositionLeftDelta [get]
```

The delta from the calibration point of the left gaze point.

8.16.3.5 GazePositionRight

```
Point GazeUtilityLibrary.DataStructs.CalibrationPoint.GazePositionRight [get], [set]
```

The right gaze point.

8.16.3.6 GazePositionRightDelta

```
Point GazeUtilityLibrary.DataStructs.CalibrationPoint.GazePositionRightDelta [get]
```

The delta from the calibration point of the right gaze point.

8.16.3.7 HasData

```
bool GazeUtilityLibrary.DataStructs.CalibrationPoint.HasData  [get], [set]
```

Flag to indicate whether data has been collected for this calibration point.

8.16.3.8 HasFailed

```
bool GazeUtilityLibrary.DataStructs.CalibrationPoint.HasFailed  [get], [set]
```

Flag to indicate whether data has been collected for this calibration point.

8.16.3.9 Index

```
int GazeUtilityLibrary.DataStructs.CalibrationPoint.Index  [get]
```

The index of the calibration point.

8.16.3.10 Position

```
Point GazeUtilityLibrary.DataStructs.CalibrationPoint.Position  [get], [set]
```

The position of the calibration point.

8.16.4 Event Documentation

8.16.4.1 PropertyChanged

```
PropertyChangedEventHandler?  GazeUtilityLibrary.DataStructs.CalibrationPoint.PropertyChanged
```

Event to trigger property changes.

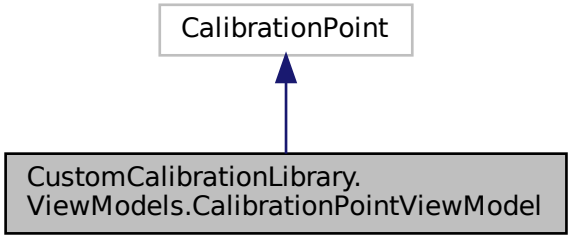
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/CalibrationPoint.cs

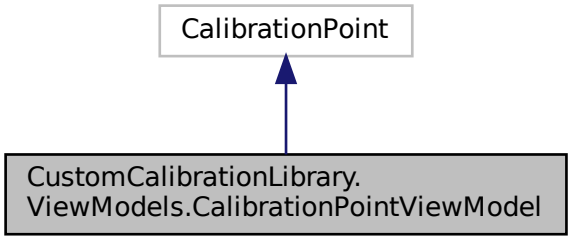
8.17 CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel Class Reference

The view model for a calibration point.

Inheritance diagram for CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel:



Collaboration diagram for CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel:



Public Member Functions

- CalibrationPointViewModel (Point point, int index)

  Initializes a new instance of the CalibrationPointViewModel class.
- CalibrationPointViewModel (CalibrationPoint point)

  Initializes a new instance of the CalibrationPointViewModel class.

Properties

- Brush? PointColor `[get]`

  The color of the calibration point.

8.17.1 Detailed Description

The view model for a calibration point.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 CalibrationPointViewModel() [1/2]

```
CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel.CalibrationPointViewModel (
            Point point,
            int index )  [inline]
```

Initializes a new instance of the CalibrationPointViewModel class.

Parameters

point	The position of the calibration point.
index	The index of the calibration point.

8.17.2.2 CalibrationPointViewModel() [2/2]

```
CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel.CalibrationPointViewModel (
            CalibrationPoint point )  [inline]
```

Initializes a new instance of the CalibrationPointViewModel class.

Parameters

point	The calibration point object.

8.17.3 Property Documentation

8.17.3.1 PointColor

```
Brush?  CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel.PointColor  [get]
```

The color of the calibration point.

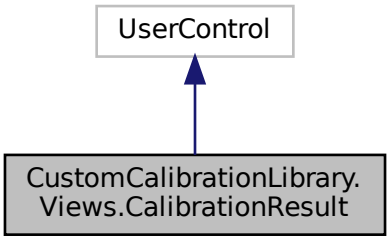
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/CalibrationPointViewModel.cs

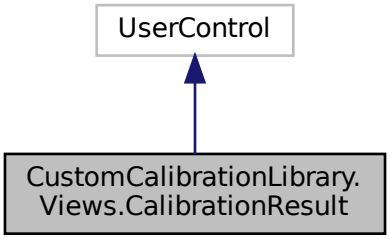
8.18 CustomCalibrationLibrary.Views.CalibrationResult Class Reference

Interaction logic for CalibrationResult.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationResult:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationResult:



Public Member Functions

- CalibrationResult (CalibrationModel model)

  Initializes a new instance of the CalibrationResult class.

8.18.1 Detailed Description

Interaction logic for CalibrationResult.xaml

8.18.2 Constructor & Destructor Documentation

8.18.2.1 CalibrationResult()

```
CustomCalibrationLibrary.Views.CalibrationResult.CalibrationResult (
            CalibrationModel model )  [inline]
```

Initializes a new instance of the CalibrationResult class.

Parameters

model	The calibration model.

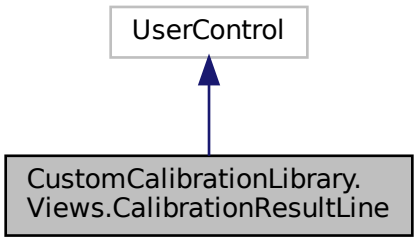
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/CalibrationResult.xaml.cs

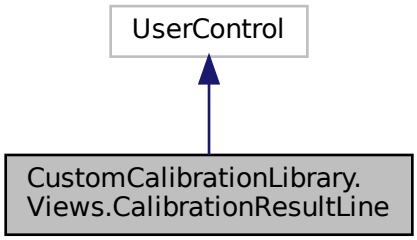
8.19 CustomCalibrationLibrary.Views.CalibrationResultLine Class Reference

Interaction logic for CalibrationResultLine.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationResultLine:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationResultLine:



8.19.1 Detailed Description

Interaction logic for CalibrationResultLine.xaml

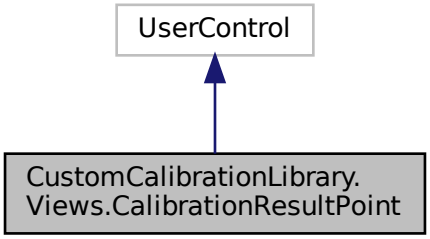
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/CalibrationResultLine.xaml.cs

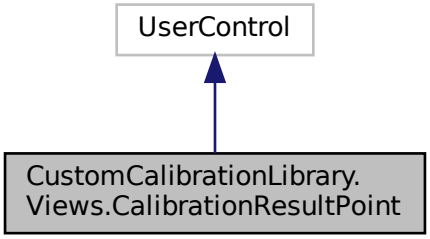
8.20 CustomCalibrationLibrary.Views.CalibrationResultPoint Class Reference

Interaction logic for CalibrationResultPoint.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationResultPoint:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationResultPoint:



Public Member Functions

- CalibrationResultPoint ()

  Initializes a new instance of the CalibrationResultPoint class.

8.20.1 Detailed Description

Interaction logic for CalibrationResultPoint.xaml

8.20.2 Constructor & Destructor Documentation

8.20.2.1 CalibrationResultPoint()

```
CustomCalibrationLibrary.Views.CalibrationResultPoint.CalibrationResultPoint ( )  [inline]
```

Initializes a new instance of the CalibrationResultPoint class.

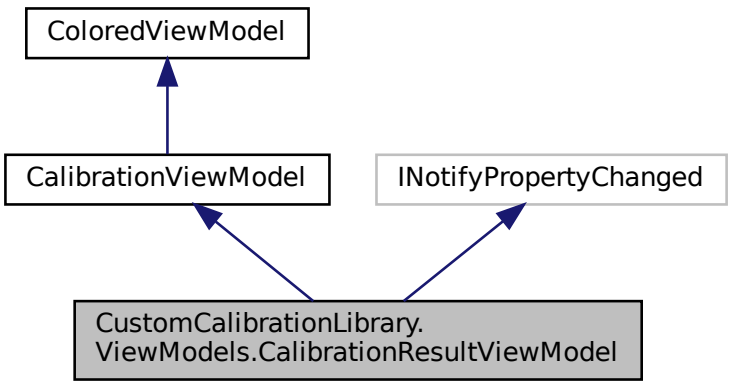
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/CalibrationResultPoint.xaml.cs

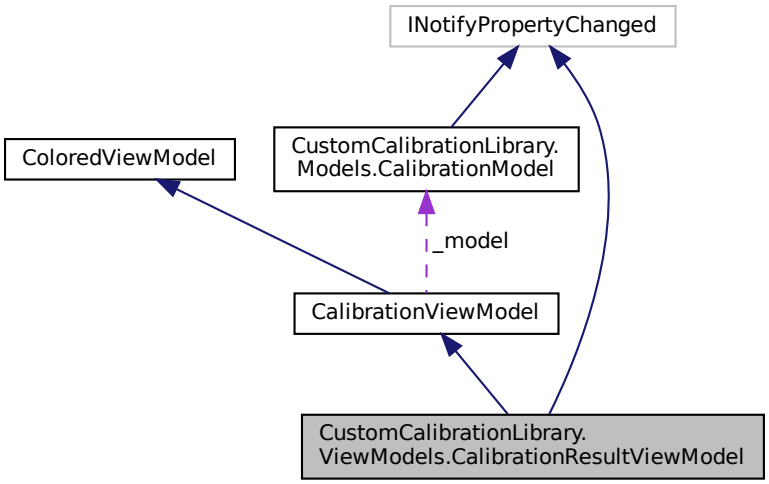
8.21 CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel Class Reference

View model class of the gaze calibration result.

Inheritance diagram for CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel:



Collaboration diagram for CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel:



Public Member Functions

- CalibrationResultViewModel (CalibrationModel model)

  Constructor
- void OnGazeToggle ()

  Toggle the visibility of the live gaze point.

Properties

- ICommand CalibrationRestartCommand `[get]`

  Command to restart the calibration
- ICommand CalibrationAcceptCommand `[get]`

  Command to accept the calibration
- ICommand GazeVisibilityCommand `[get]`

  Command to toggle the visibility of the live gaze point
- LiveGazePoint GazePoint `[get]`

  The position of the live gaze point
- double AccuracyLeft `[get]`

  The calibration accuracy of the left eye.
- double AccuracyRight `[get]`

  The calibration accuracy of the right eye.
- Visibility SuccessVisibility `[get]`

  The visibility flag for all items if the accuracy is acceptable.
- Visibility AlertVisibility `[get]`

  The visibility flag for all items if the accuracy is too low.
- Visibility RedoTimerVisibility `[get]`

  The visibility flag for all items if the accuracy is too low.
- int RemainingSec `[get, set]`

  The number or remaining seconds before an automatic calibration restart.

Events

- PropertyChangedEventHandler? PropertyChanged

  The protperty changed handler.

Additional Inherited Members

8.21.1 Detailed Description

View model class of the gaze calibration result.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 CalibrationResultViewModel()

```
CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.CalibrationResultViewModel (
            CalibrationModel model ) [inline]
```

Constructor

Parameters

model	The claibration model

8.21.3 Member Function Documentation

8.21.3.1 OnGazeToggle()

```
void CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.OnGazeToggle ( ) [inline]
```

Toggle the visibility of the live gaze point.

8.21.4 Property Documentation

8.21.4.1 AccuracyLeft

```
double CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.AccuracyLeft [get]
```

The calibration accuracy of the left eye.

8.21.4.2 AccuracyRight

```
double CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.AccuracyRight  [get]
```

The calibration accuracy of the right eye.

8.21.4.3 AlertVisibility

```
Visibility CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.AlertVisibility
[get]
```

The visibility flag for all items if the accuracy is too low.

8.21.4.4 CalibrationAcceptCommand

```
ICommand CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.CalibrationAccept↩
Command  [get]
```

Command to accept the calibration

8.21.4.5 CalibrationRestartCommand

```
ICommand CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.CalibrationRestart↩
Command  [get]
```

Command to restart the calibration

8.21.4.6 GazePoint

```
LiveGazePoint CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.GazePoint  [get]
```

The position of the live gaze point

8.21.4.7 GazeVisibilityCommand

```
ICommand CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.GazeVisibilityCommand
[get]
```

Command to toggle the visibility of the live gaze point

8.21.4.8 RedoTimerVisibility

```
Visibility CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.RedoTimerVisibility
[get]
```

The visibility flag for all items if the accuracy is too low.

8.21.4.9 RemainingSec

```
int CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.RemainingSec  [get], [set]
```

The number or remaining seconds before an automatic calibration restart.

8.21.4.10 SuccessVisibility

```
Visibility CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel.SuccessVisibility
[get]
```

The visibility flag for all items if the accuracy is acceptable.

8.21.5 Event Documentation

8.21.5.1 PropertyChanged

```
PropertyChangedEventHandler?  CustomCalibrationLibrary.ViewModels.CalibrationResultView↩
Model.PropertyChanged
```

The protperty changed handler.

The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/CalibrationResultViewModel.cs

8.22 Tobii.Research.Addons.CalibrationValidationPoint Class Reference

Represents a collected point that goes into the calibration validation. It contains calculated values for accuracy and precision as well as the original gaze samples collected for the point.

Public Member Functions

- override string ToString ()

  Convert validation values to a string.

Properties

- NormalizedPoint2D Coordinates `[get]`

  The 2D coordinates of this point (in Active Display Coordinate System).
- float AccuracyLeftEye `[get]`

  The accuracy in degrees for the left eye.
- float PrecisionLeftEye `[get]`

  The precision (standard deviation) in degrees for the left eye.
- float PrecisionRMSLeftEye `[get]`

  The precision (root mean square of sample-to-sample error) in degrees for the left eye.
- float AccuracyRightEye `[get]`

  The accuracy in degrees for the right eye.
- float PrecisionRightEye `[get]`

  The precision (standard deviation) in degrees for the right eye.
- float PrecisionRMSRightEye `[get]`

  The precision (root mean square of sample-to-sample error) in degrees for the right eye.
- bool TimedOut `[get]`

  A boolean indicating if there was a timeout while collecting data for this point.
- GazeDataEventArgs[ ] GazeData `[get]`

  The gaze data samples collected for this point. These samples are the base for the calculated accuracy and precision.

8.22.1 Detailed Description

Represents a collected point that goes into the calibration validation. It contains calculated values for accuracy and precision as well as the original gaze samples collected for the point.

8.22.2 Member Function Documentation

8.22.2.1 ToString()

```
override string Tobii.Research.Addons.CalibrationValidationPoint.ToString ( )  [inline]
```

Convert validation values to a string.

Returns

The validation string.

8.22.3 Property Documentation

8.22.3.1 AccuracyLeftEye

```
float Tobii.Research.Addons.CalibrationValidationPoint.AccuracyLeftEye [get]
```

The accuracy in degrees for the left eye.

8.22.3.2 AccuracyRightEye

```
float Tobii.Research.Addons.CalibrationValidationPoint.AccuracyRightEye [get]
```

The accuracy in degrees for the right eye.

8.22.3.3 Coordinates

```
NormalizedPoint2D Tobii.Research.Addons.CalibrationValidationPoint.Coordinates [get]
```

The 2D coordinates of this point (in Active Display Coordinate System).

8.22.3.4 GazeData

```
GazeDataEventArgs [] Tobii.Research.Addons.CalibrationValidationPoint.GazeData [get]
```

The gaze data samples collected for this point. These samples are the base for the calculated accuracy and precision.

8.22.3.5 PrecisionLeftEye

```
float Tobii.Research.Addons.CalibrationValidationPoint.PrecisionLeftEye [get]
```

The precision (standard deviation) in degrees for the left eye.

8.22.3.6 PrecisionRightEye

```
float Tobii.Research.Addons.CalibrationValidationPoint.PrecisionRightEye [get]
```

The precision (standard deviation) in degrees for the right eye.

8.22.3.7 PrecisionRMSLeftEye

```
float Tobii.Research.Addons.CalibrationValidationPoint.PrecisionRMSLeftEye [get]
```

The precision (root mean square of sample-to-sample error) in degrees for the left eye.

8.22.3.8 PrecisionRMSRightEye

```
float Tobii.Research.Addons.CalibrationValidationPoint.PrecisionRMSRightEye [get]
```

The precision (root mean square of sample-to-sample error) in degrees for the right eye.

8.22.3.9 TimedOut

```
bool Tobii.Research.Addons.CalibrationValidationPoint.TimedOut [get]
```

A boolean indicating if there was a timeout while collecting data for this point.

The documentation for this class was generated from the following file:

- source/TobiiProSdkAddons/ScreenBasedCalibrationValidation.cs

8.23 Tobii.Research.Addons.CalibrationValidationResult Class Reference

Contains the result of the calibration validation.

Public Member Functions

- override string ToString ()
  Convert validation values to a string.

Properties

- List< CalibrationValidationPoint > Points [get]
  The results of the calibration validation per point (same points as were collected).
- float AverageAccuracyLeftEye [get]
  The accuracy in degrees averaged over all collected points for the left eye.
- float AveragePrecisionLeftEye [get]
  The precision (standard deviation) in degrees averaged over all collected points for the left eye.
- float AveragePrecisionRMSLeftEye [get]
  The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the left eye.
- float AverageAccuracyRightEye [get]
  The accuracy in degrees averaged over all collected points for the right eye.
- float AveragePrecisionRightEye [get]
  The precision (standard deviation) in degrees averaged over all collected points for the right eye.
- float AveragePrecisionRMSRightEye [get]
  The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the right eye.

8.23.1 Detailed Description

Contains the result of the calibration validation.

8.23.2 Member Function Documentation

8.23.2.1 ToString()

```
override string Tobii.Research.Addons.CalibrationValidationResult.ToString ( )  [inline]
```

Convert validation values to a string.

Returns

The validation string.

8.23.3 Property Documentation

8.23.3.1 AverageAccuracyLeftEye

```
float Tobii.Research.Addons.CalibrationValidationResult.AverageAccuracyLeftEye  [get]
```

The accuracy in degrees averaged over all collected points for the left eye.

8.23.3.2 AverageAccuracyRightEye

```
float Tobii.Research.Addons.CalibrationValidationResult.AverageAccuracyRightEye  [get]
```

The accuracy in degrees averaged over all collected points for the right eye.

8.23.3.3 AveragePrecisionLeftEye

```
float Tobii.Research.Addons.CalibrationValidationResult.AveragePrecisionLeftEye  [get]
```

The precision (standard deviation) in degrees averaged over all collected points for the left eye.

8.23.3.4 AveragePrecisionRightEye

```
float Tobii.Research.Addons.CalibrationValidationResult.AveragePrecisionRightEye [get]
```

The precision (standard deviation) in degrees averaged over all collected points for the right eye.

8.23.3.5 AveragePrecisionRMSLeftEye

```
float Tobii.Research.Addons.CalibrationValidationResult.AveragePrecisionRMSLeftEye [get]
```

The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the left eye.

8.23.3.6 AveragePrecisionRMSRightEye

```
float Tobii.Research.Addons.CalibrationValidationResult.AveragePrecisionRMSRightEye [get]
```

The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the right eye.

8.23.3.7 Points

```
List<CalibrationValidationPoint> Tobii.Research.Addons.CalibrationValidationResult.Points
[get]
```

The results of the calibration validation per point (same points as were collected).

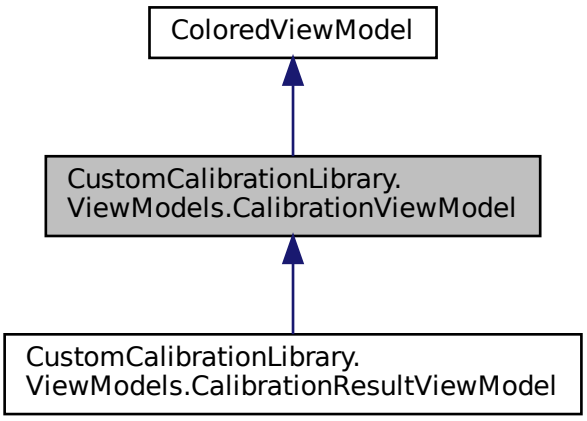
The documentation for this class was generated from the following file:

- source/TobiiProSdkAddons/ScreenBasedCalibrationValidation.cs

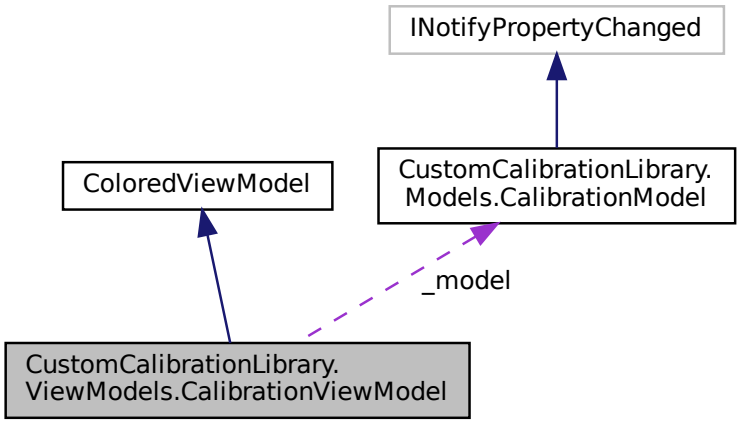
8.24 CustomCalibrationLibrary.ViewModels.CalibrationViewModel Class Reference

The view model class of the calibration view

Inheritance diagram for CustomCalibrationLibrary.ViewModels.CalibrationViewModel:



Collaboration diagram for CustomCalibrationLibrary.ViewModels.CalibrationViewModel:



Public Member Functions

- CalibrationViewModel (CalibrationModel model)

  Constructor

Protected Attributes

- CalibrationModel _model

  The claibration model.

Properties

- ObservableCollection< CalibrationPointViewModel > CalibrationPoints [get]

  The collection of calibration points to be shown on the view
- Cursor CursorType [get]

  The type of the curser to allow hiding the mouse pointer.

8.24.1 Detailed Description

The view model class of the calibration view

8.24.2 Constructor & Destructor Documentation

8.24.2.1 CalibrationViewModel()

```
CustomCalibrationLibrary.ViewModels.CalibrationViewModel.CalibrationViewModel (
            CalibrationModel model ) [inline]
```

Constructor

Parameters

model	The calibration model

8.24.3 Member Data Documentation

8.24.3.1 _model

```
CalibrationModel CustomCalibrationLibrary.ViewModels.CalibrationViewModel._model [protected]
```

The claibration model.

8.24.4 Property Documentation

8.24.4.1 CalibrationPoints

```
ObservableCollection<CalibrationPointViewModel> CustomCalibrationLibrary.ViewModels.Calibration↩
ViewModel.CalibrationPoints [get]
```

The collection of calibration points to be shown on the view

8.24.4.2 CursorType

```
Cursor CustomCalibrationLibrary.ViewModels.CalibrationViewModel.CursorType [get]
```

The type of the curser to allow hiding the mouse pointer.

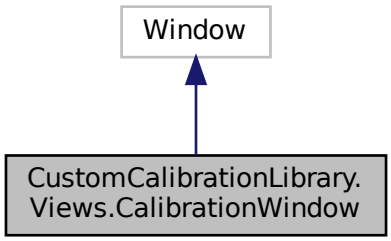
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/CalibrationViewModel.cs

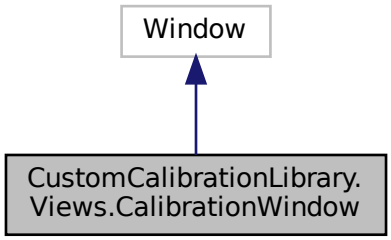
8.25 CustomCalibrationLibrary.Views.CalibrationWindow Class Reference

Interaction logic for MainWindow.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationWindow:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationWindow:



8.25.1 Detailed Description

Interaction logic for MainWindow.xaml

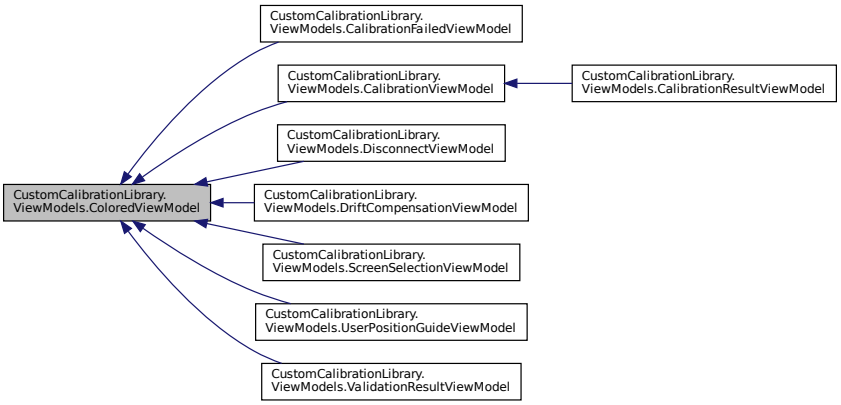
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/CalibrationWindow.xaml.cs

8.26 CustomCalibrationLibrary.ViewModels.ColoredViewModel Class Reference

The base view model for coloring the view.

Inheritance diagram for CustomCalibrationLibrary.ViewModels.ColoredViewModel:



Public Member Functions

- ColoredViewModel (Color backgroundColor, Color frameColor)

  Constructor

Properties

- Color BackgroundColor `[get]`

  The background color of the canvas.
- Color FrameColor `[get]`

  The background color of the frame.

8.26.1 Detailed Description

The base view model for coloring the view.

8.26.2 Constructor & Destructor Documentation

8.26.2.1 ColoredViewModel()

```
CustomCalibrationLibrary.ViewModels.ColoredViewModel.ColoredViewModel (
            Color backgroundColor,
            Color frameColor )  [inline]
```

Constructor

Parameters

backgroundColor	The background color of the view.
frameColor	The frame color of the view.

8.26.3 Property Documentation

8.26.3.1 BackgroundColor

```
Color CustomCalibrationLibrary.ViewModels.ColoredViewModel.BackgroundColor  [get]
```

The background color of the canvas.

8.26.3.2 FrameColor

```
Color CustomCalibrationLibrary.ViewModels.ColoredViewModel.FrameColor  [get]
```

The background color of the frame.

The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/ColoredViewModel.cs

8.27 GazeUtilityLibrary.ConfigItem Class Reference

configuration file class

Public Member Functions

- ConfigItem ()

  Initializes a new instance of the ConfigItem class.

Properties

- string CalibrationLogColumnOrder `[get, set]`

  Allows to define the order and the delimiters between the different calibration data values.
- string[ ] CalibrationLogColumnTitle `[get, set]`

  Defines the titles of the calibration data log value columns.
- bool CalibrationLogWriteOutput `[get, set]`

  Defines whether gaze calibration data is written to a log file.
- double[ ][ ] CalibrationPoints `[get, set]`

  Define the calibration points to be shown during the calibration process.
- double CalibrationAccuracyThreshold `[get, set]`

  Define the calibration accuracy threshold in degrees.
- int CalibrationRetries `[get, set]`

  The number of automatic retries if the calibration fails due to a missed CalibrationAccuracyThreshold.
- double DriftCompensationDispersionThreshold `[get, set]`

  In order to detect a fixation with the I-DT algorithm a dispersion threshold is required. Provide an angle in degrees.
- double DriftCompensationDispersionThresholdMax `[get, set]`

  In order to prevent drift compensation from getting out of hand limit the maximal allowed dispresion. If the drift compensation angle is larger than the here defined degrees, no compensation is applied. Provide an angle in degrees.
- int DriftCompensationDurationThreshold `[get, set]`

  Specifies the amount of time (in milliseconds) required to fixate the target during drift compensation.
- int DriftCompensationTimer `[get, set]`

  Specifies the amount of time (in milliseconds) to wait for a fixation point during drift compensation.
- bool DriftCompensationWindowShow `[get, set]`

  If set to true the drift compensation window is shown on the drift compensation command. Otherwise only the drift compensation process is done without showing the window.
- int ValidationDurationThreshold `[get, set]`

  Specifies the amount of time (in milliseconds) required to fixate the target during validation.
- string ValidationLogColumnOrder `[get, set]`

  Allows to define the order and the delimiters between the different validation data values.
- string[ ] ValidationLogColumnTitle `[get, set]`

  Defines the titles of the validation data log value columns.
- bool ValidationLogWriteOutput `[get, set]`

  Defines whether gaze validation data is written to a log file.
- double[ ][ ] ValidationPoints `[get, set]`

  Define the validation points to be shown during the validation process.
- int ValidationTimer `[get, set]`

  Specifies the amount of time (in milliseconds) to wait for a fixation point during validation.
- double ValidationAccuracyThreshold `[get, set]`

  Define the validation accuracy threshold in degrees.
- double ValidationPrecisionThreshold `[get, set]`

  Define the validation precision threshold in degrees.
- int ValidationRetries `[get, set]`

  The number of automatic retries if the validation fails due to a missed ValidationAccuracyThreshold.
- string DataLogColumnOrder `[get, set]`

  Allows to define the order and the delimiters between the different gaze data values.

- string[ ] DataLogColumnTitle [get, set]

  Defines the titles of the gaze data log value columns.
- int DataLogCount [get, set]

  Number of maximal allowed output data files in the output path. Oldest files are deleted first.
- string DataLogFormatDiameter [get, set]

  Allows to define the format of how the pupil diameter (in millimetres) will be logged.
- string DataLogFormatOrigin [get, set]

  Allows to define the format of how the gaze origin values (in millimetres) will be logged.
- string DataLogFormatNormalizedPoint [get, set]

  Allows to define the format of how normalized data points will be logged.
- string DataLogFormatTimeStamp [get, set]

  Allows to define the format of the timestamp.
- string DataLogFormatTimeStampRelative [get, set]

  Allows to define the format of the relative timestamp in milliseconds.
- string DataLogFormatValidation [get, set]

  Allows to define the format of the validation values.
- string DataLogPath [get, set]

  Defines the location of the output file. It must be the path to a folder (not a file).
- bool DataLogWriteOutput [get, set]

  Defines whether gaze data is written to a log file.
- bool DataLogDisabledOnStartup [get, set]

  Defines whether gaze data storing is disabled on Gaze application start.
- string? ConfigName [get, set]

  The name of the experiment.
- string? LicensePath [get, set]

  Defines the location of the license files. It must be the path to a folder (not a file).
- int ReadyTimer [get, set]

  Specifies the amount of time (in milliseconds) to wait for the eye tracker to become ready while it is in any other state.
- int TrackerDevice [get, set]

  Choose the tracker device (1: Tobii Pro SDK, 2: Mouse Tracker).
- int LoadingTimer [get, set]

  Specifies the amount of time (in milliseconds) to wait during the loading screen.
- string BackgroundColor [get, set]

  Defines the background color of the calibration and validation canvas.
- string FrameColor [get, set]

  Defines the color of the calibration and validation frames where titles and buttons are rendered.
- bool MouseControl [get, set]

  Defines whether the mouse cursor shall be controlled by the gaze of the subject during the experiment.
- bool MouseControlHide [get, set]

  Defines whether the mouse cursor shall be hidden during the experiment.
- bool MouseCalibrationHide [get, set]

  Defines whether the mouse cursor shall be hidden on the calibration window.
- string MouseStandardIconPath [get, set]

  Defines the Path to the standard mouse pointer icon.
- string TobiiApplicationPath [get, set]

  Defines the Tobii installation path. It must be the path to a folder (not a file).
- string TobiiCalibrate [get, set]

  The Tobii application to run a calibration.
- string TobiiCalibrateArguments [get, set]

  The arguments to pass to the calibration application. Use S as a placeholder for the device serial number and A as a placeholder for the device address.
- ConfigScreenArea ScreenArea [get, set]

  Hold the screen area once the config file is dumped during experimentation.

8.27.1 Detailed Description

configuration file class

8.27.2 Constructor & Destructor Documentation

8.27.2.1 ConfigItem()

```
GazeUtilityLibrary.ConfigItem.ConfigItem ( )  [inline]
```

Initializes a new instance of the ConfigItem class.

8.27.3 Property Documentation

8.27.3.1 BackgroundColor

```
string GazeUtilityLibrary.ConfigItem.BackgroundColor  [get], [set]
```

Defines the background color of the calibration and validation canvas.

8.27.3.2 CalibrationAccuracyThreshold

```
double GazeUtilityLibrary.ConfigItem.CalibrationAccuracyThreshold  [get], [set]
```

Define the calibration accuracy threshold in degrees.

8.27.3.3 CalibrationLogColumnOrder

```
string GazeUtilityLibrary.ConfigItem.CalibrationLogColumnOrder  [get], [set]
```

Allows to define the order and the delimiters between the different calibration data values.

8.27.3.4 CalibrationLogColumnTitle

```
string [] GazeUtilityLibrary.ConfigItem.CalibrationLogColumnTitle  [get], [set]
```

Defines the titles of the calibration data log value columns.

8.27.3.5 CalibrationLogWriteOutput

```
bool GazeUtilityLibrary.ConfigItem.CalibrationLogWriteOutput  [get], [set]
```

Defines whether gaze calibration data is written to a log file.

8.27.3.6 CalibrationPoints

```
double [][] GazeUtilityLibrary.ConfigItem.CalibrationPoints  [get], [set]
```

Define the calibration points to be shown during the calibration process.

8.27.3.7 CalibrationRetries

```
int GazeUtilityLibrary.ConfigItem.CalibrationRetries  [get], [set]
```

The number of automatic retries if the calibration fails due to a missed CalibrationAccuracyThreshold.

8.27.3.8 ConfigName

```
string?  GazeUtilityLibrary.ConfigItem.ConfigName  [get], [set]
```

The name of the experiment.

8.27.3.9 DataLogColumnOrder

```
string GazeUtilityLibrary.ConfigItem.DataLogColumnOrder  [get], [set]
```

Allows to define the order and the delimiters between the different gaze data values.

8.27.3.10 DataLogColumnTitle

```
string [] GazeUtilityLibrary.ConfigItem.DataLogColumnTitle  [get], [set]
```

Defines the titles of the gaze data log value columns.

8.27.3.11 DataLogCount

```
int GazeUtilityLibrary.ConfigItem.DataLogCount  [get], [set]
```

Number of maximal allowed output data files in the output path. Oldest files are deleted first.

8.27.3.12 DataLogDisabledOnStartup

```
bool GazeUtilityLibrary.ConfigItem.DataLogDisabledOnStartup  [get], [set]
```

Defines whether gaze data storing is disabled on Gaze application start.

8.27.3.13 DataLogFormatDiameter

```
string GazeUtilityLibrary.ConfigItem.DataLogFormatDiameter  [get], [set]
```

Allows to define the format of how the pupil diameter (in millimetres) will be logged.

8.27.3.14 DataLogFormatNormalizedPoint

```
string GazeUtilityLibrary.ConfigItem.DataLogFormatNormalizedPoint  [get], [set]
```

Allows to define the format of how normalized data points will be logged.

8.27.3.15 DataLogFormatOrigin

```
string GazeUtilityLibrary.ConfigItem.DataLogFormatOrigin  [get], [set]
```

Allows to define the format of how the gaze origin values (in millimetres) will be logged.

8.27.3.16 DataLogFormatTimeStamp

```
string GazeUtilityLibrary.ConfigItem.DataLogFormatTimeStamp  [get], [set]
```

Allows to define the format of the timestamp.

8.27.3.17 DataLogFormatTimeStampRelative

```
string GazeUtilityLibrary.ConfigItem.DataLogFormatTimeStampRelative  [get], [set]
```

Allows to define the format of the relative timestamp in milliseconds.

8.27.3.18 DataLogFormatValidation

```
string GazeUtilityLibrary.ConfigItem.DataLogFormatValidation  [get], [set]
```

Allows to define the format of the validation values.

8.27.3.19 DataLogPath

```
string GazeUtilityLibrary.ConfigItem.DataLogPath  [get], [set]
```

Defines the location of the output file. It must be the path to a folder (not a file).

8.27.3.20 DataLogWriteOutput

```
bool GazeUtilityLibrary.ConfigItem.DataLogWriteOutput  [get], [set]
```

Defines whether gaze data is written to a log file.

8.27.3.21 DriftCompensationDispersionThreshold

```
double GazeUtilityLibrary.ConfigItem.DriftCompensationDispersionThreshold  [get], [set]
```

In order to detect a fixation with the I-DT algorithm a dispersion threshold is required. Provide an angle in degrees.

8.27.3.22 DriftCompensationDispersionThresholdMax

```
double GazeUtilityLibrary.ConfigItem.DriftCompensationDispersionThresholdMax [get], [set]
```

In order to prevent drift compensation from getting out of hand limit the maximal allowed dispresion. If the drift compensation angle is larger than the here defined degrees, no compensation is applied. Provide an angle in degrees.

8.27.3.23 DriftCompensationDurationThreshold

```
int GazeUtilityLibrary.ConfigItem.DriftCompensationDurationThreshold [get], [set]
```

Specifies the amount of time (in milliseconds) required to fixate the target during drift compensation.

8.27.3.24 DriftCompensationTimer

```
int GazeUtilityLibrary.ConfigItem.DriftCompensationTimer [get], [set]
```

Specifies the amount of time (in milliseconds) to wait for a fixation point during drift compensation.

8.27.3.25 DriftCompensationWindowShow

```
bool GazeUtilityLibrary.ConfigItem.DriftCompensationWindowShow [get], [set]
```

If set to true the drift compensation window is shown on the drift compensation command. Otherwise only the drift compensation process is done without showing the window.

8.27.3.26 FrameColor

```
string GazeUtilityLibrary.ConfigItem.FrameColor [get], [set]
```

Defines the color of the calibration and validation frames where titles and buttons are rendered.

8.27.3.27 LicensePath

```
string? GazeUtilityLibrary.ConfigItem.LicensePath [get], [set]
```

Defines the location of the license files. It must be the path to a folder (not a file).

8.27.3.28 LoadingTimer

```
int GazeUtilityLibrary.ConfigItem.LoadingTimer  [get], [set]
```

Specifies the amount of time (in milliseconds) to wait during the loading screen.

8.27.3.29 MouseCalibrationHide

```
bool GazeUtilityLibrary.ConfigItem.MouseCalibrationHide  [get], [set]
```

Defines whether the mouse cursor shall be hidden on the calibration window.

8.27.3.30 MouseControl

```
bool GazeUtilityLibrary.ConfigItem.MouseControl  [get], [set]
```

Defines whether the mouse cursor shall be controlled by the gaze of the subject during the experiment.

8.27.3.31 MouseControlHide

```
bool GazeUtilityLibrary.ConfigItem.MouseControlHide  [get], [set]
```

Defines whether the mouse cursor shall be hidden during the experiment.

8.27.3.32 MouseStandardIconPath

```
string GazeUtilityLibrary.ConfigItem.MouseStandardIconPath  [get], [set]
```

Defines the Path to the standard mouse pointer icon.

8.27.3.33 ReadyTimer

```
int GazeUtilityLibrary.ConfigItem.ReadyTimer  [get], [set]
```

Specifies the amount of time (in milliseconds) to wait for the eye tracker to become ready while it is in any other state.

8.27.3.34 ScreenArea

```
ConfigScreenArea GazeUtilityLibrary.ConfigItem.ScreenArea  [get], [set]
```

Hold the screen area once the config file is dumped during experimentation.

8.27.3.35 TobiiApplicationPath

```
string GazeUtilityLibrary.ConfigItem.TobiiApplicationPath  [get], [set]
```

Defines the Tobii installation path. It must be the path to a folder (not a file).

8.27.3.36 TobiiCalibrate

```
string GazeUtilityLibrary.ConfigItem.TobiiCalibrate  [get], [set]
```

The Tobii application to run a calibration.

8.27.3.37 TobiiCalibrateArguments

```
string GazeUtilityLibrary.ConfigItem.TobiiCalibrateArguments  [get], [set]
```

The arguments to pass to the calibration application. Use S as a placeholder for the device serial number and A as a placeholder for the device address.

8.27.3.38 TrackerDevice

```
int GazeUtilityLibrary.ConfigItem.TrackerDevice  [get], [set]
```

Choose the tracker device (1: Tobii Pro SDK, 2: Mouse Tracker).

8.27.3.39 ValidationAccuracyThreshold

```
double GazeUtilityLibrary.ConfigItem.ValidationAccuracyThreshold  [get], [set]
```

Define the validation accuracy threshold in degrees.

8.27.3.40 ValidationDurationThreshold

```
int GazeUtilityLibrary.ConfigItem.ValidationDurationThreshold [get], [set]
```

Specifies the amount of time (in milliseconds) required to fixate the target during validation.

8.27.3.41 ValidationLogColumnOrder

```
string GazeUtilityLibrary.ConfigItem.ValidationLogColumnOrder [get], [set]
```

Allows to define the order and the delimiters between the different validation data values.

8.27.3.42 ValidationLogColumnTitle

```
string [] GazeUtilityLibrary.ConfigItem.ValidationLogColumnTitle [get], [set]
```

Defines the titles of the validation data log value columns.

8.27.3.43 ValidationLogWriteOutput

```
bool GazeUtilityLibrary.ConfigItem.ValidationLogWriteOutput [get], [set]
```

Defines whether gaze validation data is written to a log file.

8.27.3.44 ValidationPoints

```
double [][] GazeUtilityLibrary.ConfigItem.ValidationPoints [get], [set]
```

Define the validation points to be shown during the validation process.

8.27.3.45 ValidationPrecisionThreshold

```
double GazeUtilityLibrary.ConfigItem.ValidationPrecisionThreshold [get], [set]
```

Define the validation precision threshold in degrees.

8.27.3.46 ValidationRetries

```
int GazeUtilityLibrary.ConfigItem.ValidationRetries  [get], [set]
```

The number of automatic retries if the validation fails due to a missed ValidationAccuracyThreshold.

8.27.3.47 ValidationTimer

```
int GazeUtilityLibrary.ConfigItem.ValidationTimer  [get], [set]
```

Specifies the amount of time (in milliseconds) to wait for a fixation point during validation.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeConfiguration.cs

8.28 GazeUtilityLibrary.ConfigScreenArea Class Reference

The JSON structure of the screen area.

Public Member Functions

- ConfigScreenArea ()

  Initializes a new instance of the ConfigScreenArea class.
- ConfigScreenArea (ScreenArea screenArea)

  Initializes a new instance of the ConfigScreenArea class.

Properties

- double Width `[get, set]`

  The width of the screen.
- double Height `[get, set]`

  The height of the screen.
- double[ ] Center `[get, set]`

  The coordinates of the center point of the screen.
- double[ ] TopLeft `[get, set]`

  The coordinates of the top left point of the screen.
- double[ ] TopRight `[get, set]`

  The coordinates of the to right point of the screen.
- double[ ] BottomLeft `[get, set]`

  The coordinates of the bottom left point of the screen.
- double[ ] BottomRight `[get, set]`

  The coordinates of the bottom right point of the screen.

8.28.1 Detailed Description

The JSON structure of the screen area.

8.28.2 Constructor & Destructor Documentation

8.28.2.1 ConfigScreenArea() [1/2]

```
GazeUtilityLibrary.ConfigScreenArea.ConfigScreenArea ( )  [inline]
```

Initializes a new instance of the ConfigScreenArea class.

8.28.2.2 ConfigScreenArea() [2/2]

```
GazeUtilityLibrary.ConfigScreenArea.ConfigScreenArea (
            ScreenArea screenArea )  [inline]
```

Initializes a new instance of the ConfigScreenArea class.

Parameters

screenArea	A screen area object.

8.28.3 Property Documentation

8.28.3.1 BottomLeft

```
double [] GazeUtilityLibrary.ConfigScreenArea.BottomLeft  [get], [set]
```

The coordinates of the bottom left point of the screen.

8.28.3.2 BottomRight

```
double [] GazeUtilityLibrary.ConfigScreenArea.BottomRight  [get], [set]
```

The coordinates of the bottom right point of the screen.

8.28.3.3 Center

```
double [] GazeUtilityLibrary.ConfigScreenArea.Center  [get], [set]
```

The coordinates of the center point of the screen.

8.28.3.4 Height

```
double GazeUtilityLibrary.ConfigScreenArea.Height  [get], [set]
```

The height of the screen.

8.28.3.5 TopLeft

```
double [] GazeUtilityLibrary.ConfigScreenArea.TopLeft  [get], [set]
```

The coordinates of the top left point of the screen.

8.28.3.6 TopRight

```
double [] GazeUtilityLibrary.ConfigScreenArea.TopRight  [get], [set]
```

The coordinates of the to right point of the screen.

8.28.3.7 Width

```
double GazeUtilityLibrary.ConfigScreenArea.Width  [get], [set]
```

The width of the screen.

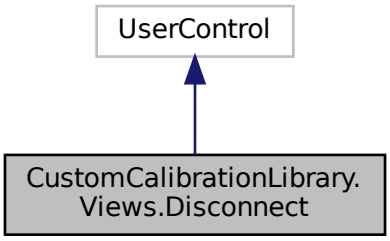
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeConfiguration.cs

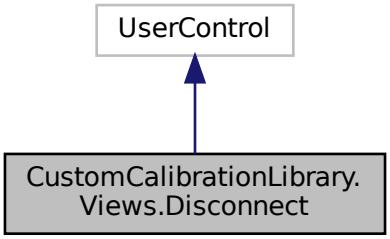
8.29 CustomCalibrationLibrary.Views.Disconnect Class Reference

Interaction logic for Disconnect.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.Disconnect:



Collaboration diagram for CustomCalibrationLibrary.Views.Disconnect:



Public Member Functions

- Disconnect (CalibrationModel model)

  Initializes a new instance of the Disconnect class.

8.29.1 Detailed Description

Interaction logic for Disconnect.xaml

8.29.2 Constructor & Destructor Documentation

8.29.2.1 Disconnect()

```
CustomCalibrationLibrary.Views.Disconnect.Disconnect (
            CalibrationModel model ) [inline]
```

Initializes a new instance of the Disconnect class.

Parameters

model	The calibration model

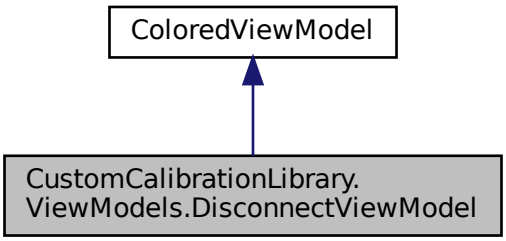
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/Disconnect.xaml.cs

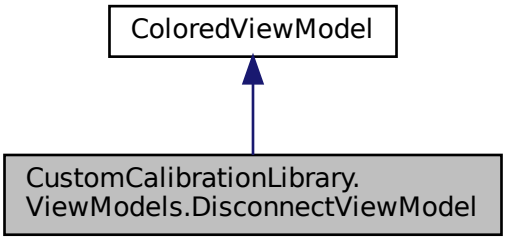
8.30 CustomCalibrationLibrary.ViewModels.DisconnectViewModel Class Reference

The view model class of the diconnect view

Inheritance diagram for CustomCalibrationLibrary.ViewModels.DisconnectViewModel:



Collaboration diagram for CustomCalibrationLibrary.ViewModels.DisconnectViewModel:



Public Member Functions

- DisconnectViewModel (CalibrationModel model)
  Constructor

Properties

- ICommand CalibrationAbortCommand `[get]`
  Command to abort the calibration

8.30.1 Detailed Description

The view model class of the diconnect view

8.30.2 Constructor & Destructor Documentation

8.30.2.1 DisconnectViewModel()

```
CustomCalibrationLibrary.ViewModels.DisconnectViewModel.DisconnectViewModel (
            CalibrationModel model ) [inline]
```

Constructor

Parameters

model	The calibration model

8.30.3 Property Documentation

8.30.3.1 CalibrationAbortCommand

```
ICommand CustomCalibrationLibrary.ViewModels.DisconnectViewModel.CalibrationAbortCommand [get]
```

Command to abort the calibration

The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/DisconnectViewModel.cs

8.31 GazeUtilityLibrary.DriftCompensation Class Reference

The class to handle drift compensation.

Public Member Functions

- DriftCompensation (Vector3 fixationPoint, int fixationFrameCount, double dispersionThreashold, double dispersionThreasholdMax)

  Initializes a new instance of the DriftCompensation class.
- void Reset ()

  Reset the drift compensation quaternion to the identity.
- void Start ()

  Start the drift compensation.
- bool Update (GazeData gazeData)

  Collect gaze data samples of a fixation and once enough samples are collected, compute the drift compensation quaternion.

Properties

- Quaternion Q `[get]`

  The drift compensation quatrenion.
- double DeviationAngle `[get]`

  The deviation angle of the drift compensation.
- double Dispersion `[get]`

  The dispersion of the drift compensation fixation.

8.31.1 Detailed Description

The class to handle drift compensation.

8.31.2 Constructor & Destructor Documentation

8.31.2.1 DriftCompensation()

```
GazeUtilityLibrary.DriftCompensation.DriftCompensation (
            Vector3 fixationPoint,
            int fixationFrameCount,
            double dispersionThreashold,
            double dispersionThreasholdMax )  [inline]
```

Initializes a new instance of the DriftCompensation class.

Parameters

fixationPoint	The target fixation point.
fixationFrameCount	The required number of frames during fixation.
dispersionThreashold	The dispersion threashold for the fixation.
dispersionThreasholdMax	The maximal allowed deviation angle.

8.31.3 Member Function Documentation

8.31.3.1 Reset()

```
void GazeUtilityLibrary.DriftCompensation.Reset ( )  [inline]
```

Reset the drift compensation quaternion to the identity.

8.31.3.2 Start()

```
void GazeUtilityLibrary.DriftCompensation.Start ( )  [inline]
```

Start the drift compensation.

8.31.3.3 Update()

```
bool GazeUtilityLibrary.DriftCompensation.Update (
            GazeData gazeData )  [inline]
```

Collect gaze data samples of a fixation and once enough samples are collected, compute the drift compensation quaternion.

Parameters

gazeData	The gaze data sample to collect if it belongs to a fixation.

Returns

True if new drift compensation is computed, false if the process is ongoing.

8.31.4 Property Documentation

8.31.4.1 DeviationAngle

```
double GazeUtilityLibrary.DriftCompensation.DeviationAngle  [get]
```

The deviation angle of the drift compensation.

8.31.4.2 Dispersion

```
double GazeUtilityLibrary.DriftCompensation.Dispersion  [get]
```

The dispersion of the drift compensation fixation.

8.31.4.3 Q

```
Quaternion GazeUtilityLibrary.DriftCompensation.Q  [get]
```

The drift compensation quatrenion.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DriftCompensation.cs

8.32 GazeUtilityLibrary.DataStructs.DriftCompensationData Class Reference

The drift compensation data structure

Public Member Functions

- DriftCompensationData (ScreenArea screen, Quaternion driftCompensation, GazeData3d gazeData)
  Constructor

Properties

- Vector2 GazePosition2d `[get]`
  The drift compensated 2d gaze position
- Vector3 GazePosition3d `[get]`
  The drift compensated 3d gaze position
- Quaternion Compensation `[get]`
  The drift compensation quaternion

8.32.1 Detailed Description

The drift compensation data structure

8.32.2 Constructor & Destructor Documentation

8.32.2.1 DriftCompensationData()

```
GazeUtilityLibrary.DataStructs.DriftCompensationData.DriftCompensationData (
            ScreenArea screen,
            Quaternion driftCompensation,
            GazeData3d gazeData )  [inline]
```

Constructor

Parameters

screen	The screen area
driftCompensation	The drift compensation quaternion
gazeData	The 3d gaze data structure

8.32.3 Property Documentation

8.32.3.1 Compensation

```
Quaternion GazeUtilityLibrary.DataStructs.DriftCompensationData.Compensation  [get]
```

The drift compensation quaternion

8.32.3.2 GazePosition2d

```
Vector2 GazeUtilityLibrary.DataStructs.DriftCompensationData.GazePosition2d  [get]
```

The drift compensated 2d gaze position

8.32.3.3 GazePosition3d

```
Vector3 GazeUtilityLibrary.DataStructs.DriftCompensationData.GazePosition3d  [get]
```

The drift compensated 3d gaze position

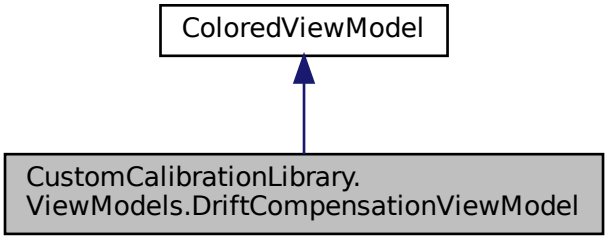
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/DriftCompensationData.cs

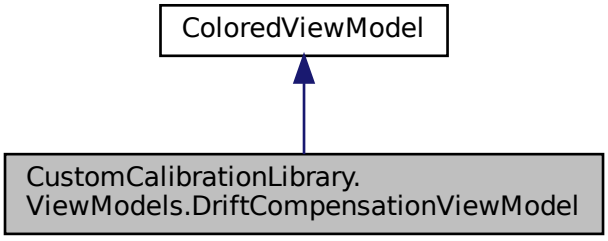
8.33 CustomCalibrationLibrary.ViewModels.DriftCompensationView↩ Model Class Reference

The view model class of the drift compensation view.

Inheritance diagram for CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel:



Collaboration diagram for CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel:



Public Member Functions

- DriftCompensationViewModel (Color backgroundColor)

  Constructor

Properties

- CalibrationPoint FixationPoint `[get, set]`

  The point on the screen which the participant is supposed to fixate.

8.33.1 Detailed Description

The view model class of the drift compensation view.

8.33.2 Constructor & Destructor Documentation

8.33.2.1 DriftCompensationViewModel()

```
CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel.DriftCompensationViewModel (
            Color backgroundColor )  [inline]
```

Constructor

8.33.3 Property Documentation

8.33.3.1 FixationPoint

```
CalibrationPoint CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel.FixationPoint
[get], [set]
```

The point on the screen which the participant is supposed to fixate.

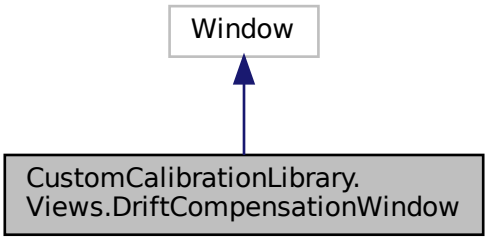
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/DriftCompensationViewModel.cs

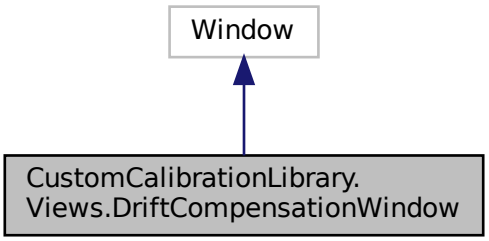
8.34 CustomCalibrationLibrary.Views.DriftCompensationWindow Class Reference

Interaction logic for DriftCompensation.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.DriftCompensationWindow:



Collaboration diagram for CustomCalibrationLibrary.Views.DriftCompensationWindow:



Public Member Functions

- DriftCompensationWindow (Color backgroundColor)

  Initializes a new instance of the DriftCompensationWindow class.

8.34.1 Detailed Description

Interaction logic for DriftCompensation.xaml

8.34.2 Constructor & Destructor Documentation

8.34.2.1 DriftCompensationWindow()

```
CustomCalibrationLibrary.Views.DriftCompensationWindow.DriftCompensationWindow (
            Color backgroundColor )  [inline]
```

Initializes a new instance of the DriftCompensationWindow class.

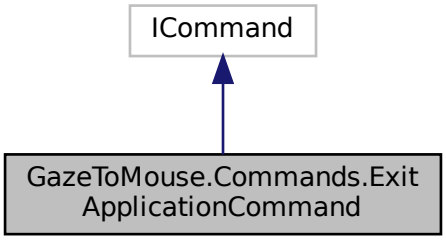
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/DriftCompensationWindow.xaml.cs

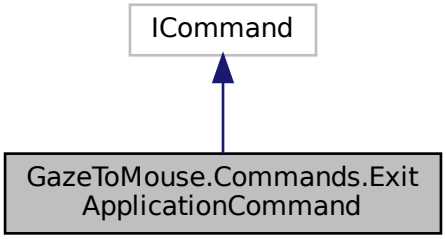
8.35 GazeToMouse.Commands.ExitApplicationCommand Class Reference

Command class to exit the application.

Inheritance diagram for GazeToMouse.Commands.ExitApplicationCommand:



Collaboration diagram for GazeToMouse.Commands.ExitApplicationCommand:



Public Member Functions

- ExitApplicationCommand (App app)

  Initializes a new instance of the ExitApplicationCommand class.
- bool CanExecute (object? parameter)

  Returns whether command can be executed or not.
- void Execute (object? parameter)

  Exit the application.

Properties

- EventHandler? CanExecuteChanged

  Event handler on can executed flag change.

8.35.1 Detailed Description

Command class to exit the application.

8.35.2 Constructor & Destructor Documentation

8.35.2.1 ExitApplicationCommand()

```
GazeToMouse.Commands.ExitApplicationCommand.ExitApplicationCommand (
            App app )  [inline]
```

Initializes a new instance of the ExitApplicationCommand class.

Parameters

app	The main application

8.35.3 Member Function Documentation

8.35.3.1 CanExecute()

```
bool GazeToMouse.Commands.ExitApplicationCommand.CanExecute (
            object?  parameter )  [inline]
```

Returns whether command can be executed or not.

Parameters

parameter	The command parameter

Returns

True

8.35.3.2 Execute()

```
void GazeToMouse.Commands.ExitApplicationCommand.Execute (
            object?  parameter )  [inline]
```

Exit the application.

Parameters

parameter	The command parameter

8.35.4 Property Documentation

8.35.4.1 CanExecuteChanged

```
EventHandler? GazeToMouse.Commands.ExitApplicationCommand.CanExecuteChanged [add], [remove]
```

Event handler on can executed flag change.

The documentation for this class was generated from the following file:

- source/GazeToMouse/Commands/ExistApplicationCommand.cs

8.36 GazeUtilityLibrary.DataStructs.EyeData Class Reference

The eye data set, including pupil information.

Public Member Functions

- EyeData (float pupilDiameter, bool isPupilDiameterValid)
  Initializes a new instance of the EyeData class.

Properties

- float PupilDiameter `[get]`
  The diameter of the pupil
- bool IsPupilDiameterValid `[get]`
  The validity flag of th epupil diameter

8.36.1 Detailed Description

The eye data set, including pupil information.

8.36.2 Constructor & Destructor Documentation

8.36.2.1 EyeData()

```
GazeUtilityLibrary.DataStructs.EyeData.EyeData (
            float pupilDiameter,
            bool isPupilDiameterValid ) [inline]
```

Initializes a new instance of the EyeData class.

Parameters

pupilDiameter	The pupil diameter.
isPupilDiameterValid	The validity of the pupil diameter.

8.36.3 Property Documentation

8.36.3.1 IsPupilDiameterValid

```
bool GazeUtilityLibrary.DataStructs.EyeData.IsPupilDiameterValid  [get]
```

The validity flag of th epupil diameter

8.36.3.2 PupilDiameter

```
float GazeUtilityLibrary.DataStructs.EyeData.PupilDiameter  [get]
```

The diameter of the pupil

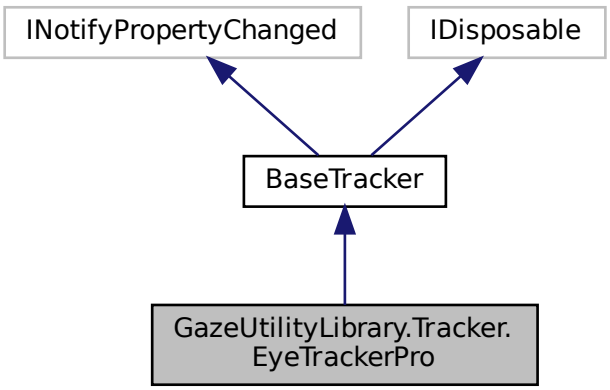
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/EyeData.cs

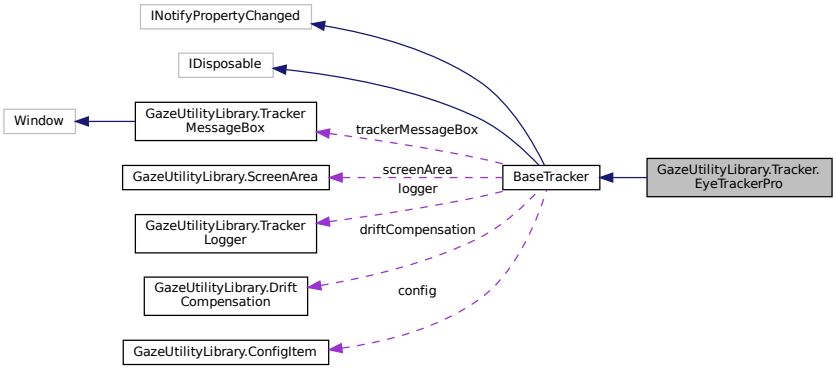
8.37 GazeUtilityLibrary.Tracker.EyeTrackerPro Class Reference

Interface to the Tobii SDK Pro engine

Inheritance diagram for GazeUtilityLibrary.Tracker.EyeTrackerPro:



Collaboration diagram for GazeUtilityLibrary.Tracker.EyeTrackerPro:



Public Member Functions

- EyeTrackerPro (TrackerLogger logger, ConfigItem config)

  Initializes a new instance of the EyeTrackerPro class.
- override async Task InitCalibrationAsync ()

  Initialise the screen based calibration.
- override void InitCalibration ()

  Initialise the screen based calibration.
- override void InitValidation ()

  Initialise the screen based calibration.
- override async Task< bool > CollectCalibrationDataAsync (Point point)

  Collects gaze data of a calibration point.
- override async Task< bool > CollectValidationDataAsync (Point point)

  Collects gaze data of a validation point.
- override async Task FinishCalibrationAsync ()

  Finish the screen based async calibration process.
- override void FinishCalibration ()

  Finish the screen based calibration process.
- override void FinishValidation ()

  Finish the screen based validation process.
- override async Task< List< GazeCalibrationData > > ApplyCalibration ()

  Compute and apply the calibration data. Transform the Tobi calibration result into the GazeCalibrationData structure.
- override? GazeValidationData ComputeValidation ()

  Compute the validation data.
- bool IsLicenseOk ()

  Determines whether the license is applied to the eyetracker device
- override bool IsInitialised ()

  Checks if the tracker device exists.
- override string PatternReplace (string pattern)

  Replaces a patten string with information from the eye tracker. Supported patterns are S for the serial number and A for the address.

Protected Member Functions

- override void InitDriftCompensation ()

  Initialise the drift compensation.
- override int GetFixationFrameCount (int durationThreshold)

  Get the number of required gaze samples to compute a fixation. This is based on the duration threshold and the sample rate of the device.
- override Vector3 GetUnitDirection ()

  Get the unit vector pointing in the direction of the gaze vector.

Additional Inherited Members

8.37.1 Detailed Description

Interface to the Tobii SDK Pro engine

See also

GazeHelper.TrackerHandler

8.37.2 Constructor & Destructor Documentation

8.37.2.1 EyeTrackerPro()

```
GazeUtilityLibrary.Tracker.EyeTrackerPro.EyeTrackerPro (
            TrackerLogger logger,
            ConfigItem config )  [inline]
```

Initializes a new instance of the EyeTrackerPro class.

Parameters

logger	The logger.
config	The config item.

8.37.3 Member Function Documentation

8.37.3.1 ApplyCalibration()

```
override async Task<List<GazeCalibrationData> > GazeUtilityLibrary.Tracker.EyeTrackerPro.↩
ApplyCalibration ( )  [inline], [virtual]
```

Compute and apply the calibration data. Transform the Tobi calibration result into the GazeCalibrationData structure.

Returns

The calibration data result wrapped by an async handler.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.2 CollectCalibrationDataAsync()

```
override async Task<bool> GazeUtilityLibrary.Tracker.EyeTrackerPro.CollectCalibrationDataAsync
(
            Point point )  [inline], [virtual]
```

Collects gaze data of a calibration point.

Parameters

point	

Returns

True on success, false on failure, wrapped by an async handler.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.3 CollectValidationDataAsync()

```
override async Task<bool> GazeUtilityLibrary.Tracker.EyeTrackerPro.CollectValidationDataAsync
(
            Point point )  [inline], [virtual]
```

Collects gaze data of a validation point.

Parameters

point	

Returns

True on success, false on failure, wrapped by an async handler.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.4 ComputeValidation()

```
override?  GazeValidationData GazeUtilityLibrary.Tracker.EyeTrackerPro.ComputeValidation ( )
[inline], [virtual]
```

Compute the validation data.

Returns

The validation data result.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.5 FinishCalibration()

```
override void GazeUtilityLibrary.Tracker.EyeTrackerPro.FinishCalibration ( )  [inline], [virtual]
```

Finish the screen based calibration process.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.6 FinishCalibrationAsync()

```
override async Task GazeUtilityLibrary.Tracker.EyeTrackerPro.FinishCalibrationAsync ( )  [inline],
[virtual]
```

Finish the screen based async calibration process.

Returns

An async handler

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.7 FinishValidation()

```
override void GazeUtilityLibrary.Tracker.EyeTrackerPro.FinishValidation ( )  [inline], [virtual]
```

Finish the screen based validation process.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.8 GetFixationFrameCount()

```
override int GazeUtilityLibrary.Tracker.EyeTrackerPro.GetFixationFrameCount (
            int durationThreshold )  [inline], [protected], [virtual]
```

Get the number of required gaze samples to compute a fixation. This is based on the duration threshold and the sample rate of the device.

Parameters

durationThreshold	The required fixation duration in milliseconds.

Returns

The number of required samples.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.9 GetUnitDirection()

```
override Vector3 GazeUtilityLibrary.Tracker.EyeTrackerPro.GetUnitDirection ( )  [inline],
[protected], [virtual]
```

Get the unit vector pointing in the direction of the gaze vector.

Returns

The unit vector pointing in the negative z direction.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.10 InitCalibration()

```
override void GazeUtilityLibrary.Tracker.EyeTrackerPro.InitCalibration ( )  [inline], [virtual]
```

Initialise the screen based calibration.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.11 InitCalibrationAsync()

```
override async Task GazeUtilityLibrary.Tracker.EyeTrackerPro.InitCalibrationAsync ( )  [inline],
[virtual]
```

Initialise the screen based calibration.

Returns

An async handler

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.12 InitDriftCompensation()

```
override void GazeUtilityLibrary.Tracker.EyeTrackerPro.InitDriftCompensation ( )  [inline],
[protected], [virtual]
```

Initialise the drift compensation.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.13 InitValidation()

```
override void GazeUtilityLibrary.Tracker.EyeTrackerPro.InitValidation ( )  [inline], [virtual]
```

Initialise the screen based calibration.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.14 IsInitialised()

```
override bool GazeUtilityLibrary.Tracker.EyeTrackerPro.IsInitialised ( )  [inline], [virtual]
```

Checks if the tracker device exists.

Returns

True if the tracker device exists, false otherwise.

Reimplemented from GazeUtilityLibrary.Tracker.BaseTracker.

8.37.3.15 IsLicenseOk()

```
bool GazeUtilityLibrary.Tracker.EyeTrackerPro.IsLicenseOk ( )  [inline]
```

Determines whether the license is applied to the eyetracker device

Returns

`true` if [is license ok]; otherwise, `false`.

8.37.3.16 PatternReplace()

```
override string GazeUtilityLibrary.Tracker.EyeTrackerPro.PatternReplace (
            string pattern )  [inline], [virtual]
```

Replaces a patten string with information from the eye tracker. Supported patterns are S for the serial number and A for the address.

Returns

The string where patterns were replaced.

Reimplemented from GazeUtilityLibrary.Tracker.BaseTracker.

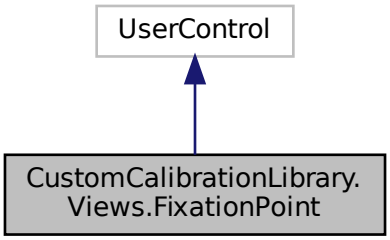
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Tracker/EyeTrackerPro.cs

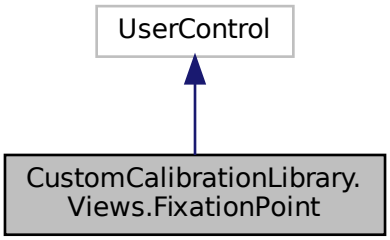
8.38 CustomCalibrationLibrary.Views.FixationPoint Class Reference

Interaction logic for FixationPoint.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.FixationPoint:



Collaboration diagram for CustomCalibrationLibrary.Views.FixationPoint:



Public Member Functions

- FixationPoint ()

  Initializes a new instance of the FixationPoint class.

8.38.1 Detailed Description

Interaction logic for FixationPoint.xaml

8.38.2 Constructor & Destructor Documentation

8.38.2.1 FixationPoint()

```
CustomCalibrationLibrary.Views.FixationPoint.FixationPoint ( )  [inline]
```

Initializes a new instance of the FixationPoint class.

The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/FixationPoint.xaml.cs

8.39 GazeUtilityLibrary.DataStructs.GazeCalibrationData Class Reference

The gaze calibration data structure

Public Member Functions

- GazeCalibrationData (double xCoord, double yCoord, double xCoordLeft, double yCoordLeft, bool validity↩ Left, double accuracyLeft, double xCoordRight, double yCoordRight, bool validityRight, double accuracy↩ Right)

  Initializes a new instance of the GazeDataArgs class.
- string[ ] Prepare (ConfigItem config)

  Prepare a list of formatted calibration data values

Properties

- double XCoord [get]

  The x coordinate of the calibration point.
- double YCoord [get]

  The y coordinate of the calibration point.
- double XCoordLeft [get]

  The x coord of the gaze point of the left eye.
- double YCoordLeft [get]

  The y coord of the gaze point of the left eye.
- bool ValidityLeft [get]

  The validity of gaze point coordinate of the left eye.
- double AccuracyLeft [get]

  The accuracy of gaze point coordinate of the left eye.
- double XCoordRight [get]

  The x coord of the gaze point of the right eye.
- double YCoordRight [get]

  The y coord of the gaze point of the right eye.
- bool ValidityRight [get]

  The validity of gaze point coordinate of the right eye.
- double AccuracyRight [get]

  The accuracy of gaze point coordinate of the right eye.

8.39.1 Detailed Description

The gaze calibration data structure

8.39.2 Constructor & Destructor Documentation

8.39.2.1 GazeCalibrationData()

```
GazeUtilityLibrary.DataStructs.GazeCalibrationData.GazeCalibrationData (
            double xCoord,
            double yCoord,
            double xCoordLeft,
            double yCoordLeft,
            bool validityLeft,
            double accuracyLeft,
            double xCoordRight,
            double yCoordRight,
            bool validityRight,
            double accuracyRight )  [inline]
```

Initializes a new instance of the GazeDataArgs class.

Parameters

xCoord	The x coord of the calibration point.
yCoord	The y coord of the calibration point.
xCoordLeft	The x coord of the gaze point of the left eye.
yCoordLeft	The y coord of the gaze point of the left eye.
validityLeft	The validity of gaze point coordinate of the left eye.
accuracyLeft	The accuracy of gaze point coordinate of the left eye.
xCoordRight	The x coord of the gaze point of the right eye.
yCoordRight	The y coord of the gaze point of the right eye.
validityRight	the validity of gaze point coordinate of the right eye.
accuracyRight	the accuracy of gaze point coordinate of the right eye.

8.39.3 Member Function Documentation

8.39.3.1 Prepare()

```
string [] GazeUtilityLibrary.DataStructs.GazeCalibrationData.Prepare (
            ConfigItem config )  [inline]
```

Prepare a list of formatted calibration data values

Parameters

config	The gaze configuration structure

Returns

A list of formatted values. Each index corresponds to a specific value. This allows to reorder the list according to a format string.

8.39.4 Property Documentation

8.39.4.1 AccuracyLeft

```
double GazeUtilityLibrary.DataStructs.GazeCalibrationData.AccuracyLeft  [get]
```

The accuracy of gaze point coordinate of the left eye.

8.39.4.2 AccuracyRight

```
double GazeUtilityLibrary.DataStructs.GazeCalibrationData.AccuracyRight [get]
```

The accuracy of gaze point coordinate of the right eye.

8.39.4.3 ValidityLeft

```
bool GazeUtilityLibrary.DataStructs.GazeCalibrationData.ValidityLeft [get]
```

The validity of gaze point coordinate of the left eye.

8.39.4.4 ValidityRight

```
bool GazeUtilityLibrary.DataStructs.GazeCalibrationData.ValidityRight [get]
```

The validity of gaze point coordinate of the right eye.

8.39.4.5 XCoord

```
double GazeUtilityLibrary.DataStructs.GazeCalibrationData.XCoord [get]
```

The x coordinate of the calibration point.

8.39.4.6 XCoordLeft

```
double GazeUtilityLibrary.DataStructs.GazeCalibrationData.XCoordLeft [get]
```

The x coord of the gaze point of the left eye.

8.39.4.7 XCoordRight

```
double GazeUtilityLibrary.DataStructs.GazeCalibrationData.XCoordRight [get]
```

The x coord of the gaze point of the right eye.

8.39.4.8 YCoord

```
double GazeUtilityLibrary.DataStructs.GazeCalibrationData.YCoord  [get]
```

The y coordinate of the calibration point.

8.39.4.9 YCoordLeft

```
double GazeUtilityLibrary.DataStructs.GazeCalibrationData.YCoordLeft  [get]
```

The y coord of the gaze point of the left eye.

8.39.4.10 YCoordRight

```
double GazeUtilityLibrary.DataStructs.GazeCalibrationData.YCoordRight  [get]
```

The y coord of the gaze point of the right eye.

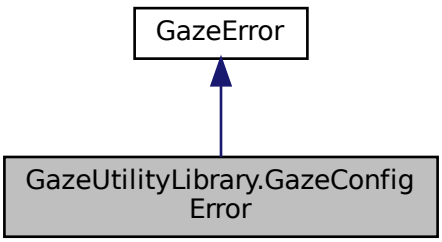
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/GazeCalibrationData.cs

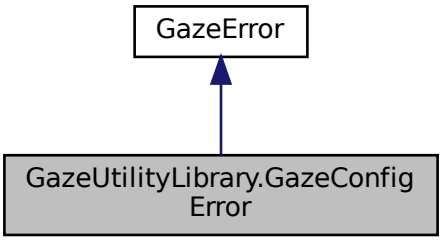
8.40 GazeUtilityLibrary.GazeConfigError Class Reference

The gaze config error class to convert error flags to binary strings.

Inheritance diagram for GazeUtilityLibrary.GazeConfigError:



Collaboration diagram for GazeUtilityLibrary.GazeConfigError:



Public Member Functions

- string GetGazeConfigErrorString ()
  Gets the gaze error string.

Properties

- EGazeConfigError Error `[set]`
  The error flags.

Additional Inherited Members

8.40.1 Detailed Description

The gaze config error class to convert error flags to binary strings.

8.40.2 Member Function Documentation

8.40.2.1 GetGazeConfigErrorString()

```
string GazeUtilityLibrary.GazeConfigError.GetGazeConfigErrorString ( )  [inline]
```

Gets the gaze error string.

Returns

the error string with binary error values if errors ocurred, the empty srting otherwise

8.40.3 Property Documentation

8.40.3.1 Error

```
EGazeConfigError GazeUtilityLibrary.GazeConfigError.Error  [set]
```

The error flags.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeError.cs

8.41 GazeUtilityLibrary.GazeConfiguration Class Reference

The gaze configuration handler.

Public Member Functions

- GazeConfiguration (TrackerLogger logger)

  Initializes a new instance of the GazeConfiguration class.
- bool InitConfig ()

  Initialise the gaze configuration by parsing and checking the configuration file.
- bool CleanupGazeOutputFile (string error)

  Close the gaze outputfile and rename it by appending error codes.
- bool CleanupCalibrationOutputFile (string error)

  Close the calibration outputfile and rename it by appending error codes.
- bool CleanupValidationOutputFile (string error)

  Close the validation outputfile and rename it by appending error codes.
- bool DumpCurrentConfigurationFile ()

  Dump current configuration to the disk.
- bool PrepareGazeOutputFile (string? subjectCode, string? outputPath)

  Prepare the gaze output file based on the configuration.
- bool PrepareCalibrationOutputFile (string? subjectCode)

  Prepare the calibration output file based on the configuration.
- bool PrepareValidationOutputFile (string? subjectCode)

  Prepare the validation output file based on the configuration.
- void WriteToGazeOutput (string[ ] formatted_values)

  Write to the gaze output file
- void WriteToCalibrationOutput (string[ ] formatted_values)

  Write to the calibration output file
- void WriteToValidationOutput (string[ ] formatted_values)

  Write to the calibration output file

Properties

- ConfigItem?? Config [get]

  The JSON structure holding the configuratyion options.

8.41.1 Detailed Description

The gaze configuration handler.

8.41.2 Constructor & Destructor Documentation

8.41.2.1 GazeConfiguration()

```
GazeUtilityLibrary.GazeConfiguration.GazeConfiguration (
            TrackerLogger logger ) [inline]
```

Initializes a new instance of the GazeConfiguration class.

Parameters

logger	The log handler.

8.41.3 Member Function Documentation

8.41.3.1 CleanupCalibrationOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.CleanupCalibrationOutputFile (
            string error ) [inline]
```

Close the calibration outputfile and rename it by appending error codes.

Parameters

error	

Returns

True on success, False on failure.

8.41.3.2 CleanupGazeOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.CleanupGazeOutputFile (
            string error )  [inline]
```

Close the gaze outputfile and rename it by appending error codes.

Parameters

error	

Returns

True on success, False on failure.

8.41.3.3 CleanupValidationOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.CleanupValidationOutputFile (
            string error )  [inline]
```

Close the validation outputfile and rename it by appending error codes.

Parameters

error	

Returns

True on success, False on failure.

8.41.3.4 DumpCurrentConfigurationFile()

```
bool GazeUtilityLibrary.GazeConfiguration.DumpCurrentConfigurationFile ( )  [inline]
```

Dump current configuration to the disk.

Returns

True on success, False on failure.

8.41.3.5 InitConfig()

```
bool GazeUtilityLibrary.GazeConfiguration.InitConfig ( )  [inline]
```

Initialise the gaze configuration by parsing and checking the configuration file.

Returns

True on success, False on failure.

8.41.3.6 PrepareCalibrationOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.PrepareCalibrationOutputFile (
            string?  subjectCode )  [inline]
```

Prepare the calibration output file based on the configuration.

Parameters

subjectCode	An optional subject code to be appended to the file name if set.

Returns

True on success, False on failure.

8.41.3.7 PrepareGazeOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.PrepareGazeOutputFile (
            string?  subjectCode,
            string?  outputPath )  [inline]
```

Prepare the gaze output file based on the configuration.

Parameters

subjectCode	An optional subject code to be appended to the file name if set.
outputPath	An optional output path where the file will be stored.

Returns

True on success, False on failure.

8.41.3.8 PrepareValidationOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.PrepareValidationOutputFile (
            string?  subjectCode )  [inline]
```

Prepare the validation output file based on the configuration.

Parameters

subjectCode	An optional subject code to be appended to the file name if set.

Returns

True on success, False on failure.

8.41.3.9 WriteToCalibrationOutput()

```
void GazeUtilityLibrary.GazeConfiguration.WriteToCalibrationOutput (
            string[] formatted_values )  [inline]
```

Write to the calibration output file

Parameters

formatted_values	The list of formatted values to be written to the file.

8.41.3.10 WriteToGazeOutput()

```
void GazeUtilityLibrary.GazeConfiguration.WriteToGazeOutput (
            string[] formatted_values )  [inline]
```

Write to the gaze output file

Parameters

formatted_values	The list of formatted values to be written to the file.

8.41.3.11 WriteToValidationOutput()

```
void GazeUtilityLibrary.GazeConfiguration.WriteToValidationOutput (
            string[] formatted_values )  [inline]
```

Write to the calibration output file

Parameters

formatted_values	The list of formatted values to be written to the file.

8.41.4 Property Documentation

8.41.4.1 Config

```
ConfigItem?? GazeUtilityLibrary.GazeConfiguration.Config [get]
```

The JSON structure holding the configuratyion options.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeConfiguration.cs

8.42 GazeUtilityLibrary.DataStructs.GazeData Class Reference

The class definition of a gaze data set

Public Member Functions

- GazeData (TimeSpan timestamp, TimeSpan timestampReceived, Vector2 gazePoint2d, bool isGaze↩ Point2dValid)

  Initializes a new instance of the GazeDataArgs class.
- GazeData (TimeSpan timestamp, TimeSpan timestampReceived, Vector2 gazePoint2dLeft, bool isGaze↩ Point2dValidLeft, Vector2 gazePoint2dRight, bool isGazePoint2dValidRight)

  Initializes a new instance of the GazeDataArgs class.
- GazeData (TimeSpan timestamp, TimeSpan timestampReceived, Vector2 gazePoint2dLeft, bool isGaze↩ Point2dValidLeft, Vector2 gazePoint2dRight, bool isGazePoint2dValidRight, Vector3 gazePoint3dLeft, bool isGazePoint3dValidLeft, Vector3 gazePoint3dRight, bool isGazePoint3dValidRight, Vector3 gazeOrigin3d↩ Left, bool isGazeOrigin3dValidLeft, Vector3 gazeOrigin3dRight, bool isGazeOrigin3dValidRight, float pupil↩ DiameterLeft, bool isPupilDiameterValidLeft, float pupilDiameterRight, bool isPupilDiameterValidRight)

  Initializes a new instance of the GazeDataArgs class.
- string[ ] Prepare (ConfigItem config, int trialId, string tag, TimeSpan startTime)

  Prepare a list of formatted gaze data values

Properties

- TimeSpan Timestamp [get]

  The timestamp of the data sample.
- TimeSpan TimestampReceived [get]

  The device timestamp of the data sample.
- GazeDataCollection? Left [get]

  The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data of the left eye.
- GazeDataCollection? Right [get]

  The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data of the right eye.
- GazeDataCollection Combined [get]

  The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data of the combined eyes.
- DriftCompensationData? DriftCompensation [get, set]

  The drift compensation information.

8.42.1 Detailed Description

The class definition of a gaze data set

8.42.2 Constructor & Destructor Documentation

8.42.2.1 GazeData() [1/3]

```
GazeUtilityLibrary.DataStructs.GazeData.GazeData (
            TimeSpan timestamp,
            TimeSpan timestampReceived,
            Vector2 gazePoint2d,
            bool isGazePoint2dValid )  [inline]
```

Initializes a new instance of the GazeDataArgs class.

Parameters

timestamp	The timestamp when the data was captured by the device.
timestampReceived	The timestamp when the data was received by the system.
gazePoint2d	The 2d coordinates of the combined gaze point.
isGazePoint2dValid	The validity of the combined 2d gaze point.

8.42.2.2 GazeData() [2/3]

```
GazeUtilityLibrary.DataStructs.GazeData.GazeData (
            TimeSpan timestamp,
```

```
            TimeSpan timestampReceived,
            Vector2 gazePoint2dLeft,
            bool isGazePoint2dValidLeft,
            Vector2 gazePoint2dRight,
            bool isGazePoint2dValidRight )  [inline]
```

Initializes a new instance of the GazeDataArgs class.

Parameters

timestamp	The timestamp when the data was captured by the device.
timestampReceived	The timestamp when the data was received by the system.
gazePoint2dLeft	The 2d coordinates of the left gaze point.
isGazePoint2dValidLeft	The validity of the left 2d gaze point.
gazePoint2dRight	The 2d coordinates of the right gaze point.
isGazePoint2dValidRight	The validity of the right 2d gaze point.

8.42.2.3 GazeData() [3/3]

```
GazeUtilityLibrary.DataStructs.GazeData.GazeData (
            TimeSpan timestamp,
            TimeSpan timestampReceived,
            Vector2 gazePoint2dLeft,
            bool isGazePoint2dValidLeft,
            Vector2 gazePoint2dRight,
            bool isGazePoint2dValidRight,
            Vector3 gazePoint3dLeft,
            bool isGazePoint3dValidLeft,
            Vector3 gazePoint3dRight,
            bool isGazePoint3dValidRight,
            Vector3 gazeOrigin3dLeft,
            bool isGazeOrigin3dValidLeft,
            Vector3 gazeOrigin3dRight,
            bool isGazeOrigin3dValidRight,
            float pupilDiameterLeft,
            bool isPupilDiameterValidLeft,
            float pupilDiameterRight,
            bool isPupilDiameterValidRight )  [inline]
```

Initializes a new instance of the GazeDataArgs class.

Parameters

timestamp	The timestamp when the data was captured by the device.
timestampReceived	The timestamp when the data was received by the system.
gazePoint2dLeft	The 2d coordinates of the left gaze point.
isGazePoint2dValidLeft	The validity of the left 2d gaze point.
gazePoint2dRight	The 2d coordinates of the right gaze point.
isGazePoint2dValidRight	The validity of the right 2d gaze point.
gazePoint3dLeft	The 3d coordinates of the left gaze point.
isGazePoint3dValidLeft	The validity of the left 3d gaze point.

Parameters

gazePoint3dRight	The 3d coordinates of the right gaze point.
isGazePoint3dValidRight	The validity of the right 3d gaze point.
gazeOrigin3dLeft	The 3d coordinates of the left gaze origin.
isGazeOrigin3dValidLeft	The validity of the left 3d gaze origin.
gazeOrigin3dRight	The 3d coordinates of the right gaze origin.
isGazeOrigin3dValidRight	The validity of the right 3d gaze origin.
pupilDiameterLeft	The pupil diameter the left eye.
isPupilDiameterValidLeft	The validity of the left pupil diameter.
pupilDiameterRight	The pupil diameter the left eye.
isPupilDiameterValidRight	The validity of the left pupil diameter.

8.42.3 Member Function Documentation

8.42.3.1 Prepare()

```
string [] GazeUtilityLibrary.DataStructs.GazeData.Prepare (
            ConfigItem config,
            int trialId,
            string tag,
            TimeSpan startTime )  [inline]
```

Prepare a list of formatted gaze data values

Parameters

config	The gaze configuration structure
trialId	The ID of the current trial.
tag	An arbitrary tag to associate with the data sample.
startTime	The system time to use toi compute the relative timestamp

Returns

A list of formatted values. Each index corresponds to a specific value. This allows to reorder the list according to a format string.

8.42.4 Property Documentation

8.42.4.1 Combined

```
GazeDataCollection GazeUtilityLibrary.DataStructs.GazeData.Combined  [get]
```

The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data of the combined eyes.

8.42.4.2 DriftCompensation

```
DriftCompensationData?  GazeUtilityLibrary.DataStructs.GazeData.DriftCompensation  [get], [set]
```

The drift compensation information.

8.42.4.3 Left

```
GazeDataCollection?  GazeUtilityLibrary.DataStructs.GazeData.Left  [get]
```

The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data of the left eye.

8.42.4.4 Right

```
GazeDataCollection?  GazeUtilityLibrary.DataStructs.GazeData.Right  [get]
```

The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data of the right eye.

8.42.4.5 Timestamp

```
TimeSpan GazeUtilityLibrary.DataStructs.GazeData.Timestamp  [get]
```

The timestamp of the data sample.

8.42.4.6 TimestampReceived

```
TimeSpan GazeUtilityLibrary.DataStructs.GazeData.TimestampReceived  [get]
```

The device timestamp of the data sample.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/GazeData.cs

8.43 GazeUtilityLibrary.DataStructs.GazeData2d Class Reference

The 2d gaze data set.

Public Member Functions

- GazeData2d (Vector2 gazePoint, bool isGazePointValid)

  Initializes a new instance of the GazeData2d class.

Properties

- Vector2 GazePoint [get]

  The 2d gaze point.
- bool IsGazePointValid [get]

  The validity flag of the 2d gaze point.

8.43.1 Detailed Description

The 2d gaze data set.

8.43.2 Constructor & Destructor Documentation

8.43.2.1 GazeData2d()

```
GazeUtilityLibrary.DataStructs.GazeData2d.GazeData2d (
            Vector2 gazePoint,
            bool isGazePointValid )  [inline]
```

Initializes a new instance of the GazeData2d class.

Parameters

gazePoint	The 2d coordinates of the gaze point.
isGazePointValid	The validity of the 2d gaze point.

8.43.3 Property Documentation

8.43.3.1 GazePoint

```
Vector2 GazeUtilityLibrary.DataStructs.GazeData2d.GazePoint  [get]
```

The 2d gaze point.

8.43.3.2 IsGazePointValid

```
bool GazeUtilityLibrary.DataStructs.GazeData2d.IsGazePointValid [get]
```

The validity flag of the 2d gaze point.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/GazeData2d.cs

8.44 GazeUtilityLibrary.DataStructs.GazeData3d Class Reference

The 3d gaze data set.

Public Member Functions

- GazeData3d (Vector3 gazePoint, bool isGazePointValid, Vector3 gazeOrigin, bool isGazeOriginValid)
  Initializes a new instance of the GazeData3d class.

Properties

- Vector3 GazePoint `[get]`
  The 3d gaze point.
- bool IsGazePointValid `[get]`
  The validity of the 3d gaze point.
- Vector3 GazeOrigin `[get]`
  The 3d origin of the gaze.
- Vector3 GazeDirection `[get]`
  The 3d gaze direction vector.
- float GazeDistance `[get]`
  The gaze distance from the origin to the gaze point.
- bool IsGazeOriginValid `[get]`
  The validity of the 3d origin.

8.44.1 Detailed Description

The 3d gaze data set.

8.44.2 Constructor & Destructor Documentation

8.44.2.1 GazeData3d()

```
GazeUtilityLibrary.DataStructs.GazeData3d.GazeData3d (
            Vector3 gazePoint,
            bool isGazePointValid,
            Vector3 gazeOrigin,
            bool isGazeOriginValid ) [inline]
```

Initializes a new instance of the GazeData3d class.

Parameters

gazePoint	The 3d coordinates of the gaze point.
isGazePointValid	The validity of the 3d gaze point.
gazeOrigin	The 3d coordinates of the gaze origin.
isGazeOriginValid	The validity of the 3d gaze origin.

8.44.3 Property Documentation

8.44.3.1 GazeDirection

```
Vector3 GazeUtilityLibrary.DataStructs.GazeData3d.GazeDirection  [get]
```

The 3d gaze direction vector.

8.44.3.2 GazeDistance

```
float GazeUtilityLibrary.DataStructs.GazeData3d.GazeDistance  [get]
```

The gaze distance from the origin to the gaze point.

8.44.3.3 GazeOrigin

```
Vector3 GazeUtilityLibrary.DataStructs.GazeData3d.GazeOrigin  [get]
```

The 3d origin of the gaze.

8.44.3.4 GazePoint

```
Vector3 GazeUtilityLibrary.DataStructs.GazeData3d.GazePoint  [get]
```

The 3d gaze point.

8.44.3.5 IsGazeOriginValid

```
bool GazeUtilityLibrary.DataStructs.GazeData3d.IsGazeOriginValid [get]
```

The validity of the 3d origin.

8.44.3.6 IsGazePointValid

```
bool GazeUtilityLibrary.DataStructs.GazeData3d.IsGazePointValid [get]
```

The validity of the 3d gaze point.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/GazeData3d.cs

8.45 GazeUtilityLibrary.DataStructs.GazeDataCollection Class Reference

The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data.

Public Member Functions

- GazeDataCollection (Vector2 gazePoint2d, bool isGazePoint2dValid)

  Initializes a new instance of the GazeDataItem class.
- GazeDataCollection (Vector2 gazePoint2d, bool isGazePoint2dValid, Vector3 gazePoint3d, bool isGaze↩ Point3dValid, Vector3 gazeOrigin3d, bool isGazeOrigin3dValid, float pupilDiameter, bool isPupilDiameter↩ Valid)

  Initializes a new instance of the GazeDataItem class.

Properties

- GazeData2d GazeData2d `[get]`

  The 2d gaze data.
- GazeData3d? GazeData3d `[get]`

  The 3d gaze data.
- EyeData? EyeData `[get]`

  Pupil data of the eye.

8.45.1 Detailed Description

The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data.

8.45.2 Constructor & Destructor Documentation

8.45.2.1 GazeDataCollection() [1/2]

```
GazeUtilityLibrary.DataStructs.GazeDataCollection.GazeDataCollection (
            Vector2 gazePoint2d,
            bool isGazePoint2dValid ) [inline]
```

Initializes a new instance of the GazeDataItem class.

Parameters

gazePoint2d	The 2d coordinates of the gaze point.
isGazePoint2dValid	The validity of the 2d gaze point.

8.45.2.2 GazeDataCollection() [2/2]

```
GazeUtilityLibrary.DataStructs.GazeDataCollection.GazeDataCollection (
            Vector2 gazePoint2d,
            bool isGazePoint2dValid,
            Vector3 gazePoint3d,
            bool isGazePoint3dValid,
            Vector3 gazeOrigin3d,
            bool isGazeOrigin3dValid,
            float pupilDiameter,
            bool isPupilDiameterValid )  [inline]
```

Initializes a new instance of the GazeDataItem class.

Parameters

gazePoint2d	The 2d coordinates of the gaze point.
isGazePoint2dValid	The validity of the 2d gaze point.
gazePoint3d	The 3d coordinates of the gaze point.
isGazePoint3dValid	The validity of the 3d gaze point.
gazeOrigin3d	The 3d coordinates of the gaze origin.
isGazeOrigin3dValid	The validity of the 3d gaze origin.
pupilDiameter	The pupil diameter.
isPupilDiameterValid	The validity of the pupil diameter.

8.45.3 Property Documentation

8.45.3.1 EyeData

```
EyeData?  GazeUtilityLibrary.DataStructs.GazeDataCollection.EyeData  [get]
```

Pupil data of the eye.

8.45.3.2 GazeData2d

```
GazeData2d GazeUtilityLibrary.DataStructs.GazeDataCollection.GazeData2d  [get]
```

The 2d gaze data.

8.45.3.3 GazeData3d

```
GazeData3d?  GazeUtilityLibrary.DataStructs.GazeDataCollection.GazeData3d  [get]
```

The 3d gaze data.

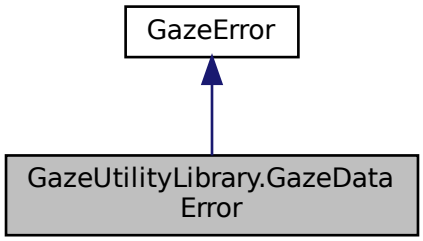
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/GazeDataCollection.cs

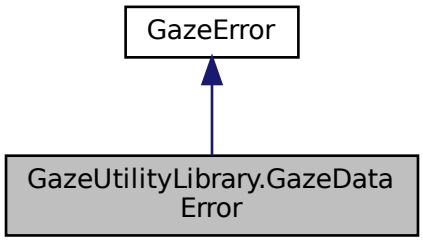
8.46 GazeUtilityLibrary.GazeDataError Class Reference

The gaze data error class to convert error flags to binary strings.

Inheritance diagram for GazeUtilityLibrary.GazeDataError:



Collaboration diagram for GazeUtilityLibrary.GazeDataError:



Public Member Functions

- string GetGazeDataErrorString ()

  Gets the gaze error string.

Properties

- EGazeDataError Error [set]

  The error flags.

Additional Inherited Members

8.46.1 Detailed Description

The gaze data error class to convert error flags to binary strings.

8.46.2 Member Function Documentation

8.46.2.1 GetGazeDataErrorString()

```
string GazeUtilityLibrary.GazeDataError.GetGazeDataErrorString ( )  [inline]
```

Gets the gaze error string.

Returns

the error string with binary error values if errors ocurred, the empty srting otherwise

8.46.3 Property Documentation

8.46.3.1 Error

```
EGazeDataError GazeUtilityLibrary.GazeDataError.Error  [set]
```

The error flags.

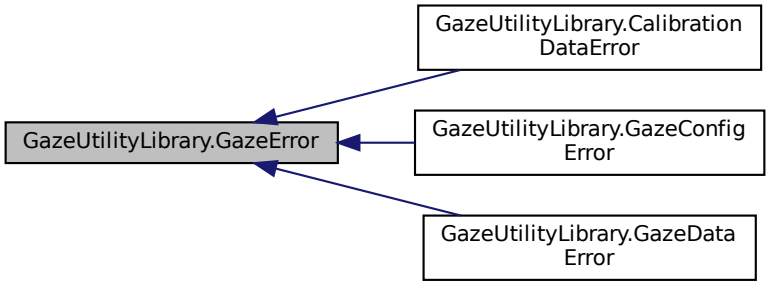
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeError.cs

8.47 GazeUtilityLibrary.GazeError Class Reference

The base error class to convert error flags to binary strings.

Inheritance diagram for GazeUtilityLibrary.GazeError:



Protected Member Functions

- string ConvertToBinString (int val, int len)

  Converts a integer value to a binary string.

8.47.1 Detailed Description

The base error class to convert error flags to binary strings.

8.47.2 Member Function Documentation

8.47.2.1 ConvertToBinString()

```
string GazeUtilityLibrary.GazeError.ConvertToBinString (
            int val,
            int len )  [inline], [protected]
```

Converts a integer value to a binary string.

Parameters

val	The value.
len	The length of the binary string.

Returns

a binary string of specified length, left-padded with '0'

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeError.cs

8.48 GazeUtilityLibrary.DataStructs.GazeValidationData Class Reference

The gaze validation data structure

Public Member Functions

- GazeValidationData ()

  Initializes a new instance of the GazeValidationData class.
- GazeValidationData (float accuracyLeft, float accuracyRight, float precisionLeft, float precisionRight, float precisionRmsLeft, float precisionRmsRight)

  Initializes a new instance of the GazeValidationData class.
- void AddPoint (Vector2 point, float accuracyLeft, float accuracyRight, float precisionLeft, float precisionRight, float precisionRmsLeft, float precisionRmsRight)

  Add a new validation point to the list.

Properties

- float AccuracyLeft `[get]`

  The accuracy in degrees averaged over all collected points for the left eye.
- float AccuracyRight `[get]`

  The accuracy in degrees averaged over all collected points for the right eye.
- float PrecisionLeft `[get]`

  The precision (standard deviation) in degrees averaged over all collected points for the left eye.
- float PrecisionRight `[get]`

  The precision (standard deviation) in degrees averaged over all collected points for the right eye.
- float PrecisionRmsLeft `[get]`

  The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the left eye.
- float PrecisionRmsRight `[get]`

  The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the right eye.
- List< GazeValidationPoint > Points `[get]`

  The list of all

8.48.1 Detailed Description

The gaze validation data structure

8.48.2 Constructor & Destructor Documentation

8.48.2.1 GazeValidationData() [1/2]

```
GazeUtilityLibrary.DataStructs.GazeValidationData.GazeValidationData ( )  [inline]
```

Initializes a new instance of the GazeValidationData class.

8.48.2.2 GazeValidationData() [2/2]

```
GazeUtilityLibrary.DataStructs.GazeValidationData.GazeValidationData (
            float accuracyLeft,
            float accuracyRight,
            float precisionLeft,
            float precisionRight,
            float precisionRmsLeft,
            float precisionRmsRight )  [inline]
```

Initializes a new instance of the GazeValidationData class.

Parameters

accuracyLeft	The accuracy in degrees averaged over all collected points for the left eye.
accuracyRight	The accuracy in degrees averaged over all collected points for the right eye.
precisionLeft	The precision (standard deviation) in degrees averaged over all collected points for the left eye.
precisionRight	The precision (standard deviation) in degrees averaged over all collected points for the right eye.
precisionRmsLeft	The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the left eye.
precisionRmsRight	The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the right eye.

8.48.3 Member Function Documentation

8.48.3.1 AddPoint()

```
void GazeUtilityLibrary.DataStructs.GazeValidationData.AddPoint (
            Vector2 point,
            float accuracyLeft,
            float accuracyRight,
```

```
            float precisionLeft,
            float precisionRight,
            float precisionRmsLeft,
            float precisionRmsRight )  [inline]
```

Add a new validation point to the list.

Parameters

point	The validation point coordinates.
accuracyLeft	The accuracy in degrees averaged over all collected points for the left eye.
accuracyRight	The accuracy in degrees averaged over all collected points for the right eye.
precisionLeft	The precision (standard deviation) in degrees averaged over all collected points for the left eye.
precisionRight	The precision (standard deviation) in degrees averaged over all collected points for the right eye.
precisionRmsLeft	The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the left eye.
precisionRmsRight	The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the right eye.

8.48.4 Property Documentation

8.48.4.1 AccuracyLeft

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.AccuracyLeft  [get]
```

The accuracy in degrees averaged over all collected points for the left eye.

8.48.4.2 AccuracyRight

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.AccuracyRight  [get]
```

The accuracy in degrees averaged over all collected points for the right eye.

8.48.4.3 Points

```
List<GazeValidationPoint> GazeUtilityLibrary.DataStructs.GazeValidationData.Points  [get]
```

The list of all

8.48.4.4 PrecisionLeft

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.PrecisionLeft [get]
```

The precision (standard deviation) in degrees averaged over all collected points for the left eye.

8.48.4.5 PrecisionRight

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.PrecisionRight [get]
```

The precision (standard deviation) in degrees averaged over all collected points for the right eye.

8.48.4.6 PrecisionRmsLeft

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.PrecisionRmsLeft [get]
```

The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the left eye.

8.48.4.7 PrecisionRmsRight

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.PrecisionRmsRight [get]
```

The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the right eye.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/GazeValidationData.cs

8.49 GazeUtilityLibrary.DataStructs.GazeValidationPoint Class Reference

A validation point.

Public Member Functions

- GazeValidationPoint (Vector2 point, GazeValidationData result)

  Initializes a new instance of the GazeValidationPoint class.
- string[ ] Prepare (ConfigItem config)

  Prepare a list of formatted calibration data values

Properties

- Vector2 Point [get]

  The validation point.
- GazeValidationData Result [get]

  The validation result of this point.

8.49.1 Detailed Description

A validation point.

8.49.2 Constructor & Destructor Documentation

8.49.2.1 GazeValidationPoint()

```
GazeUtilityLibrary.DataStructs.GazeValidationPoint.GazeValidationPoint (
            Vector2 point,
            GazeValidationData result ) [inline]
```

Initializes a new instance of the GazeValidationPoint class.

Parameters

point	The validation point.
result	The validation result of this point.

8.49.3 Member Function Documentation

8.49.3.1 Prepare()

```
string [] GazeUtilityLibrary.DataStructs.GazeValidationPoint.Prepare (
            ConfigItem config ) [inline]
```

Prepare a list of formatted calibration data values

Parameters

config	The gaze configuration structure

Returns

A list of formatted values. Each index corresponds to a specific value. This allows to reorder the list according to a format string.

8.49.4 Property Documentation

8.49.4.1 Point

```
Vector2 GazeUtilityLibrary.DataStructs.GazeValidationPoint.Point  [get]
```

The validation point.

8.49.4.2 Result

```
GazeValidationData GazeUtilityLibrary.DataStructs.GazeValidationPoint.Result  [get]
```

The validation result of this point.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/GazeValidationData.cs

8.50 GazeUtilityLibrary.JsonConfigParser Class Reference

The config file "config.json" is parsed and its values are attributed to the ConfigItem class.

Public Member Functions

- JsonConfigParser (TrackerLogger logger)
  Initializes a new instance of the JsonConfigParser class.
- ConfigItem? ParseJsonConfig (ref GazeConfigError error)
  Parses the json configuration.
- void SerializeJsonConfig (ConfigItem item, string path)
  Serializes the json configuration object to a string and writes it to a file.
- ConfigItem GetDefaultConfig ()
  Gets the default configuration values.

8.50.1 Detailed Description

The config file "config.json" is parsed and its values are attributed to the ConfigItem class.

8.50.2 Constructor & Destructor Documentation

8.50.2.1 JsonConfigParser()

```
GazeUtilityLibrary.JsonConfigParser.JsonConfigParser (
            TrackerLogger logger )  [inline]
```

Initializes a new instance of the JsonConfigParser class.

Parameters

logger	The logger.

8.50.3 Member Function Documentation

8.50.3.1 GetDefaultConfig()

```
ConfigItem GazeUtilityLibrary.JsonConfigParser.GetDefaultConfig ( )  [inline]
```

Gets the default configuration values.

Returns

the default configuration values.

8.50.3.2 ParseJsonConfig()

```
ConfigItem?  GazeUtilityLibrary.JsonConfigParser.ParseJsonConfig (
            ref GazeConfigError error )  [inline]
```

Parses the json configuration.

Returns

the updated ConfigItem class.

8.50.3.3 SerializeJsonConfig()

```
void GazeUtilityLibrary.JsonConfigParser.SerializeJsonConfig (
            ConfigItem item,
            string path )  [inline]
```

Serializes the json configuration object to a string and writes it to a file.

Parameters

item	The json configuration item.
path	The path where the file will be written.

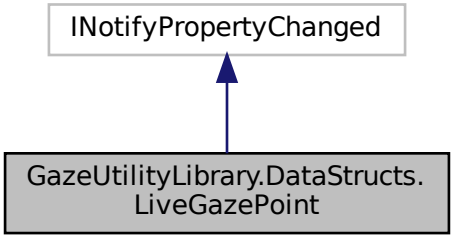
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeConfiguration.cs

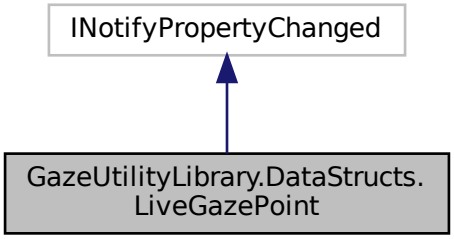
8.51 GazeUtilityLibrary.DataStructs.LiveGazePoint Class Reference

The live gaze point used for verification during the calibration process.

Inheritance diagram for GazeUtilityLibrary.DataStructs.LiveGazePoint:



Collaboration diagram for GazeUtilityLibrary.DataStructs.LiveGazePoint:



Properties

- double X `[get, set]`

  The normalized x coordinate on the screen
- double Y `[get, set]`

  The normalized y coordinate on the screen
- bool Visibility `[get, set]`

  The visiblity flag.

Events

- PropertyChangedEventHandler? PropertyChanged
  Event to trigger property changes.

8.51.1 Detailed Description

The live gaze point used for verification during the calibration process.

8.51.2 Property Documentation

8.51.2.1 Visibility

```
bool GazeUtilityLibrary.DataStructs.LiveGazePoint.Visibility  [get], [set]
```

The visiblity flag.

8.51.2.2 X

```
double GazeUtilityLibrary.DataStructs.LiveGazePoint.X  [get], [set]
```

The normalized x coordinate on the screen

8.51.2.3 Y

```
double GazeUtilityLibrary.DataStructs.LiveGazePoint.Y  [get], [set]
```

The normalized y coordinate on the screen

8.51.3 Event Documentation

8.51.3.1 PropertyChanged

```
PropertyChangedEventHandler?  GazeUtilityLibrary.DataStructs.LiveGazePoint.PropertyChanged
```

Event to trigger property changes.

The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/LiveGazePoint.cs

8.52 CustomCalibrationLibrary.ViewModels.Monitor Class Reference

A representation of the screen.

Public Member Functions

- Monitor (int index, string name)

  Initializes a new instance of the Monitor class.

Properties

- string Name `[get]`

  The name of the screen.
- int Index `[get]`

  The screen index.

8.52.1 Detailed Description

A representation of the screen.

8.52.2 Constructor & Destructor Documentation

8.52.2.1 Monitor()

```
CustomCalibrationLibrary.ViewModels.Monitor.Monitor (
            int index,
            string name )  [inline]
```

Initializes a new instance of the Monitor class.

Parameters

index	The screen index.
name	The name of the screen.

8.52.3 Property Documentation

8.52.3.1 Index

```
int CustomCalibrationLibrary.ViewModels.Monitor.Index  [get]
```

The screen index.

8.52.3.2 Name

```
string CustomCalibrationLibrary.ViewModels.Monitor.Name  [get]
```

The name of the screen.

The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/ScreenSelectionViewModel.cs

8.53 GazeUtilityLibrary.MouseHider Class Reference

hide standard mouse pointer and resore it

Public Member Functions

- MouseHider (TrackerLogger logger)
  Initializes a new instance of the MouseHider class.
- void HideCursor ()
  Hides the cursor.
- void ShowCursor (string? pathToCur)
  Shows the cursor.

8.53.1 Detailed Description

hide standard mouse pointer and resore it

8.53.2 Constructor & Destructor Documentation

8.53.2.1 MouseHider()

```
GazeUtilityLibrary.MouseHider.MouseHider (
            TrackerLogger logger )  [inline]
```

Initializes a new instance of the MouseHider class.

Parameters

logger	The logger.

8.53.3 Member Function Documentation

8.53.3.1 HideCursor()

```
void GazeUtilityLibrary.MouseHider.HideCursor ( )  [inline]
```

Hides the cursor.

Hides the standard mouse pointer by replacing the current icon with a transparent icon.

8.53.3.2 ShowCursor()

```
void GazeUtilityLibrary.MouseHider.ShowCursor (
            string?  pathToCur )  [inline]
```

Shows the cursor.

the standard mouse pointer by replacing the current icon with the standard mouse pointer icon

Parameters

pathToCur	The path to the standard mouse pointer icon.

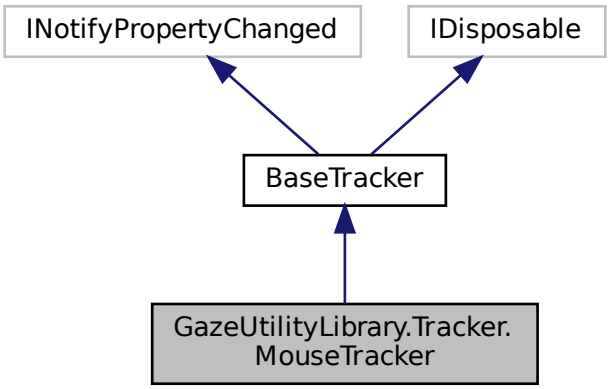
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/MouseHider.cs

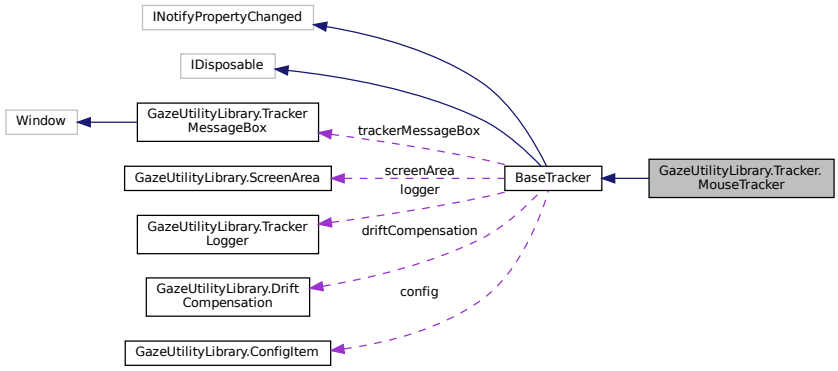
8.54 GazeUtilityLibrary.Tracker.MouseTracker Class Reference

This class is used to hook into the system mouse events and track the position

Inheritance diagram for GazeUtilityLibrary.Tracker.MouseTracker:



Collaboration diagram for GazeUtilityLibrary.Tracker.MouseTracker:



Public Member Functions

- MouseTracker (TrackerLogger logger, ConfigItem config)

  Initializes a new instance of the MouseTracker class.
- override Task< List< GazeCalibrationData > > ApplyCalibration ()

  Apply the calibration data. This is device specific and must be overwritten by the device class.
- void Start ()

  Hooks the callback function HookCallback(int, IntPtr, IntPtr) to mouse events.
- void Stop ()

  Removes to mouse event hook.
- override Task InitCalibrationAsync ()

  Initialise the async calibartion process. This is device specific and must be overwritten by the device class.
- override void InitValidation ()

Initialise the validation process. This is device specific and must be overwritten by the device class.

- override Task FinishCalibrationAsync ()

  Finish the async calibartion process. This is device specific and must be overwritten by the device class.
- override void FinishValidation ()

  Finish the validation process. This is device specific and must be overwritten by the device class.
- override Task< bool > CollectCalibrationDataAsync (Point point)

  Collect calibration data on a calibration point. This is device specific and must be overwritten by the device class.
- override Task< bool > CollectValidationDataAsync (Point point)

  Collect validation data on a validation point. This is device specific and must be overwritten by the device class.
- override void InitCalibration ()

  Initialise the calibartion process. This is device specific and must be overwritten by the device class.
- override void FinishCalibration ()

  Finish the calibartion process. This is device specific and must be overwritten by the device class.
- override? GazeValidationData ComputeValidation ()

  Apply the validation data. This is device specific and must be overwritten by the device class.

Protected Member Functions

- override void Dispose (bool disposing)

  Releases unmanaged and - optionally - managed resources.
- override int GetFixationFrameCount (int durationThreshold)

  Get the number of required gaze samples to compute a fixation. This is device specific and must be overwritten by the device because the duration of fixation point detection depends on the frame rate of the device.
- override Vector3 GetUnitDirection ()

  Get the unit vector pointing in the direction of the gaze vector. This is device specific as the gaze data are represented in a coordinate system as defined by the device.
- override void InitDriftCompensation ()

  Initialise the drift compensation. This is device specific and must be overwritten by the device class.

Additional Inherited Members

8.54.1 Detailed Description

This class is used to hook into the system mouse events and track the position

See also

GazeHelper.TrackerHandler

8.54.2 Constructor & Destructor Documentation

8.54.2.1 MouseTracker()

```
GazeUtilityLibrary.Tracker.MouseTracker.MouseTracker (
            TrackerLogger logger,
            ConfigItem config )  [inline]
```

Initializes a new instance of the MouseTracker class.

Parameters

logger	The logger.
config	The config item.

8.54.3 Member Function Documentation

8.54.3.1 ApplyCalibration()

```
override Task<List<GazeCalibrationData> > GazeUtilityLibrary.Tracker.MouseTracker.Apply↩
Calibration ( )  [inline], [virtual]
```

Apply the calibration data. This is device specific and must be overwritten by the device class.

Returns

The calibration data result wrapped by an async handler.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.2 CollectCalibrationDataAsync()

```
override Task<bool> GazeUtilityLibrary.Tracker.MouseTracker.CollectCalibrationDataAsync (
            Point point )  [inline], [virtual]
```

Collect calibration data on a calibration point. This is device specific and must be overwritten by the device class.

Parameters

point	The calibration point for which to collect data

Returns

True on success, false on failure, wrapped by an async handler.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.3 CollectValidationDataAsync()

```
override Task<bool> GazeUtilityLibrary.Tracker.MouseTracker.CollectValidationDataAsync (
            Point point )  [inline], [virtual]
```

Collect validation data on a validation point. This is device specific and must be overwritten by the device class.

Parameters

point	The calibration point for which to collect data

Returns

True on success, false on failure, wrapped by an async handler.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.4 ComputeValidation()

```
override? GazeValidationData GazeUtilityLibrary.Tracker.MouseTracker.ComputeValidation ( )
[inline], [virtual]
```

Apply the validation data. This is device specific and must be overwritten by the device class.

Returns

The validation data result.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.5 Dispose()

```
override void GazeUtilityLibrary.Tracker.MouseTracker.Dispose (
            bool disposing ) [inline], [protected], [virtual]
```

Releases unmanaged and - optionally - managed resources.

Parameters

disposing	`true` to release both managed and unmanaged resources; `false` to release only unmanaged resources.

Reimplemented from GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.6 FinishCalibration()

```
override void GazeUtilityLibrary.Tracker.MouseTracker.FinishCalibration ( ) [inline], [virtual]
```

Finish the calibartion process. This is device specific and must be overwritten by the device class.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.7 FinishCalibrationAsync()

```
override Task GazeUtilityLibrary.Tracker.MouseTracker.FinishCalibrationAsync ( )  [inline],
[virtual]
```

Finish the async calibartion process. This is device specific and must be overwritten by the device class.

Returns

An async handler

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.8 FinishValidation()

```
override void GazeUtilityLibrary.Tracker.MouseTracker.FinishValidation ( )  [inline], [virtual]
```

Finish the validation process. This is device specific and must be overwritten by the device class.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.9 GetFixationFrameCount()

```
override int GazeUtilityLibrary.Tracker.MouseTracker.GetFixationFrameCount (
            int durationThreshold )  [inline], [protected], [virtual]
```

Get the number of required gaze samples to compute a fixation. This is device specific and must be overwritten by the device because the duration of fixation point detection depends on the frame rate of the device.

Parameters

durationThreshold	The required fixation duration in milliseconds.

Returns

The number of gaze samples to require for fixation detection.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.10 GetUnitDirection()

```
override Vector3 GazeUtilityLibrary.Tracker.MouseTracker.GetUnitDirection ( )  [inline], [protected],
[virtual]
```

Get the unit vector pointing in the direction of the gaze vector. This is device specific as the gaze data are represented in a coordinate system as defined by the device.

Returns

The unit vector

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.11 InitCalibration()

```
override void GazeUtilityLibrary.Tracker.MouseTracker.InitCalibration ( )  [inline], [virtual]
```

Initialise the calibartion process. This is device specific and must be overwritten by the device class.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.12 InitCalibrationAsync()

```
override Task GazeUtilityLibrary.Tracker.MouseTracker.InitCalibrationAsync ( )  [inline],
[virtual]
```

Initialise the async calibartion process. This is device specific and must be overwritten by the device class.

Returns

An async handler

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.13 InitDriftCompensation()

```
override void GazeUtilityLibrary.Tracker.MouseTracker.InitDriftCompensation ( )  [inline],
[protected], [virtual]
```

Initialise the drift compensation. This is device specific and must be overwritten by the device class.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.14 InitValidation()

```
override void GazeUtilityLibrary.Tracker.MouseTracker.InitValidation ( )  [inline], [virtual]
```

Initialise the validation process. This is device specific and must be overwritten by the device class.

Implements GazeUtilityLibrary.Tracker.BaseTracker.

8.54.3.15 Start()

```
void GazeUtilityLibrary.Tracker.MouseTracker.Start ( )  [inline]
```

Hooks the callback function HookCallback(int, IntPtr, IntPtr) to mouse events.

8.54.3.16 Stop()

```
void GazeUtilityLibrary.Tracker.MouseTracker.Stop ( )  [inline]
```

Removes to mouse event hook.

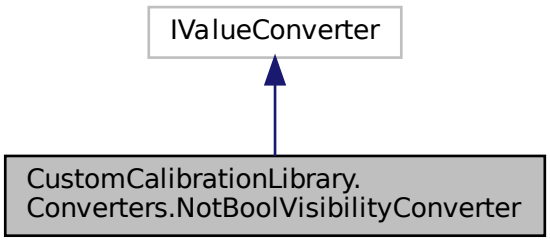
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Tracker/MouseTracker.cs

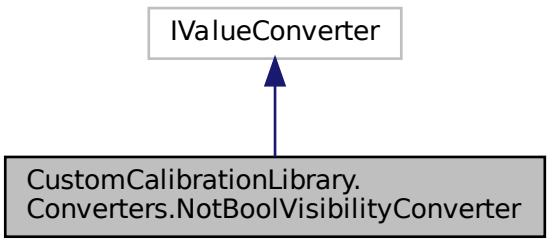
8.55 CustomCalibrationLibrary.Converters.NotBoolVisibilityConverter Class Reference

Converts True to Hidden and False to Visible

Inheritance diagram for CustomCalibrationLibrary.Converters.NotBoolVisibilityConverter:



Collaboration diagram for CustomCalibrationLibrary.Converters.NotBoolVisibilityConverter:



Public Member Functions

- object Convert (object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)

  Value converter.
- object ConvertBack (object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)

  Reverted value converter.

8.55.1 Detailed Description

Converts True to Hidden and False to Visible

8.55.2 Member Function Documentation

8.55.2.1 Convert()

```
object CustomCalibrationLibrary.Converters.NotBoolVisibilityConverter.Convert (
            object value,
            Type targetType,
            object parameter,
            System.Globalization.CultureInfo culture )  [inline]
```

Value converter.

Parameters

value	The value to convert.
targetType	The type of the target value.
parameter	The conversion parameter.
culture	The language localisation.

Returns

The converted value object

Exceptions

InvalidOperationException	

8.55.2.2 ConvertBack()

```
object CustomCalibrationLibrary.Converters.NotBoolVisibilityConverter.ConvertBack (
            object value,
```

```
Type targetType,
object parameter,
System.Globalization.CultureInfo culture )  [inline]
```

Reverted value converter.

Parameters

value	The value to convert.
targetType	The type of the target value.
parameter	The conversion parameter.
culture	The language localisation.

Returns

The converted value object

Exceptions

NotSupportedException	

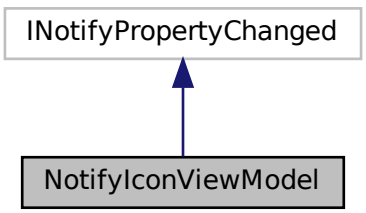
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Converters/NotBoolVisibilityConverter.cs

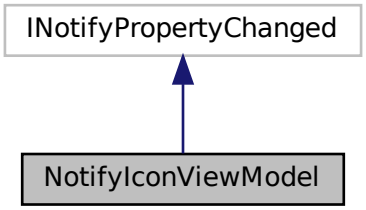
8.56 NotifyIconViewModel Class Reference

Provides bindable properties and commands for the NotifyIcon.

Inheritance diagram for NotifyIconViewModel:



Collaboration diagram for NotifyIconViewModel:



Public Member Functions

- NotifyIconViewModel (App app)

  The constructor.

Properties

- double DriftDeviationAngle `[get, set]`

  The deviation angle of the currently active drift compensation.
- ICommand ExitApplicationCommand `[get]`

  Command to exit the application
- ICommand StartCalibrationCommand `[get]`

  Command to start the calibration
- ICommand StartValidationCommand `[get]`

  Command to start the validation
- ICommand StartDriftCompensationCommand `[get]`

  Command to start the drift compensation
- ICommand ResetDriftCompensationCommand `[get]`

  Command to reset the drift compensation
- ICommand UpdateDriftDeviationAngleCommand `[get]`

  Command to update the drift deviation angle value

Events

- PropertyChangedEventHandler? PropertyChanged

  The protperty changed handler.

8.56.1 Detailed Description

Provides bindable properties and commands for the NotifyIcon.

8.56.2 Constructor & Destructor Documentation

8.56.2.1 NotifyIconViewModel()

```
NotifyIconViewModel.NotifyIconViewModel (
            App app )  [inline]
```

The constructor.

Parameters

app	The main application.

8.56.3 Property Documentation

8.56.3.1 DriftDeviationAngle

```
double NotifyIconViewModel.DriftDeviationAngle  [get], [set]
```

The deviation angle of the currently active drift compensation.

8.56.3.2 ExitApplicationCommand

```
ICommand NotifyIconViewModel.ExitApplicationCommand  [get]
```

Command to exit the application

8.56.3.3 ResetDriftCompensationCommand

```
ICommand NotifyIconViewModel.ResetDriftCompensationCommand  [get]
```

Command to reset the drift compensation

8.56.3.4 StartCalibrationCommand

```
ICommand NotifyIconViewModel.StartCalibrationCommand [get]
```

Command to start the calibration

8.56.3.5 StartDriftCompensationCommand

```
ICommand NotifyIconViewModel.StartDriftCompensationCommand [get]
```

Command to start the drift compensation

8.56.3.6 StartValidationCommand

```
ICommand NotifyIconViewModel.StartValidationCommand [get]
```

Command to start the validation

8.56.3.7 UpdateDriftDeviationAngleCommand

```
ICommand NotifyIconViewModel.UpdateDriftDeviationAngleCommand [get]
```

Command to update the drift deviation angle value

8.56.4 Event Documentation

8.56.4.1 PropertyChanged

```
PropertyChangedEventHandler? NotifyIconViewModel.PropertyChanged
```

The protperty changed handler.

The documentation for this class was generated from the following file:

- source/GazeToMouse/NotifyIconViewModel.cs

8.57 GazeControlLibrary.PipeCommand Class Reference

The JSON structure of a pipe command.

Public Member Functions

- PipeCommand (string command, bool reset, int? trialId, string label)

  Initializes a new instance of the PipeCommand class.

Properties

- string Command `[get, set]`

  The optional pipe command to be sent.
- string Label `[get, set]`

  An optional label to annotate gaze data.
- int? TrialId `[get, set]`

  An optional trial ID to annotate gaze data.
- bool? ResetStartTime `[get, set]`

  An optional flag to indicate whether the relative timestamp should be reset.

8.57.1 Detailed Description

The JSON structure of a pipe command.

8.57.2 Constructor & Destructor Documentation

8.57.2.1 PipeCommand()

```
GazeControlLibrary.PipeCommand.PipeCommand (
            string command,
            bool reset,
            int?  trialId,
            string label )  [inline]
```

Initializes a new instance of the PipeCommand class.

Parameters

command	The pipe command to be sent.
reset	A flag to indicate whether the relative timestamp should be reset.
trialId	An optional trial ID to annotate gaze data.
label	An optional label to annotate gaze data.

8.57.3 Property Documentation

8.57.3.1 Command

```
string GazeControlLibrary.PipeCommand.Command [get], [set]
```

The optional pipe command to be sent.

8.57.3.2 Label

```
string GazeControlLibrary.PipeCommand.Label [get], [set]
```

An optional label to annotate gaze data.

8.57.3.3 ResetStartTime

```
bool? GazeControlLibrary.PipeCommand.ResetStartTime [get], [set]
```

An optional flag to indicate whether the relative timestamp should be reset.

8.57.3.4 TrialId

```
int? GazeControlLibrary.PipeCommand.TrialId [get], [set]
```

An optional trial ID to annotate gaze data.

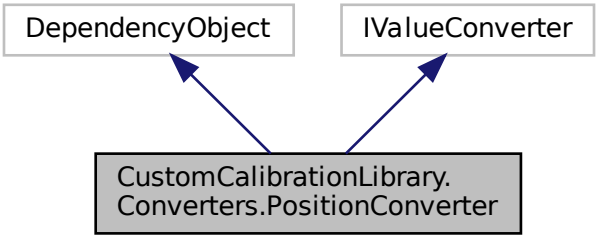
The documentation for this class was generated from the following file:

- source/GazeControlLibrary/PipeCommand.cs

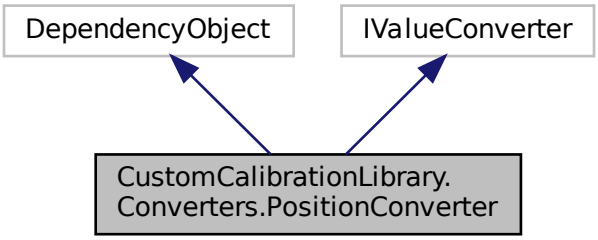
8.58 CustomCalibrationLibrary.Converters.PositionConverter Class Reference

Converter class to convert a normalized coordinate to a pixel coordinate.

Inheritance diagram for CustomCalibrationLibrary.Converters.PositionConverter:



Collaboration diagram for CustomCalibrationLibrary.Converters.PositionConverter:



Public Member Functions

- object Convert (object value, Type targetType, object parameter, CultureInfo culture)

  Value converter.
- object ConvertBack (object value, Type targetType, object parameter, CultureInfo culture)

  Reverted value converter.

Static Public Attributes

- static readonly DependencyProperty OffsetProperty

  The custom offset property of the value converter.

Properties

- string?? Offset `[get, set]`

  The position offset.

8.58.1 Detailed Description

Converter class to convert a normalized coordinate to a pixel coordinate.

8.58.2 Member Function Documentation

8.58.2.1 Convert()

```
object CustomCalibrationLibrary.Converters.PositionConverter.Convert (
            object value,
            Type targetType,
            object parameter,
            CultureInfo culture )  [inline]
```

Value converter.

Parameters

value	The value to convert.
targetType	The type of the target value.
parameter	The conversion parameter.
culture	The language localisation.

Returns

The converted value object

8.58.2.2 ConvertBack()

```
object CustomCalibrationLibrary.Converters.PositionConverter.ConvertBack (
            object value,
            Type targetType,
            object parameter,
            CultureInfo culture )  [inline]
```

Reverted value converter.

Parameters

value	The value to convert.
targetType	The type of the target value.
parameter	The conversion parameter.
culture	The language localisation.

Returns

The converted value object

Exceptions

NotSupportedException	

8.58.3 Member Data Documentation

8.58.3.1 OffsetProperty

```
readonly DependencyProperty CustomCalibrationLibrary.Converters.PositionConverter.Offset↩
Property  [static]
```

Initial value:

```
=
        DependencyProperty.Register("Offset", typeof(string), typeof(PositionConverter), new
       PropertyMetadata(null))
```

The custom offset property of the value converter.

8.58.4 Property Documentation

8.58.4.1 Offset

```
string??  CustomCalibrationLibrary.Converters.PositionConverter.Offset  [get], [set]
```

The position offset.

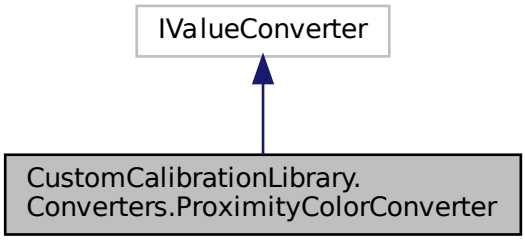
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Converters/PositionConverter.cs

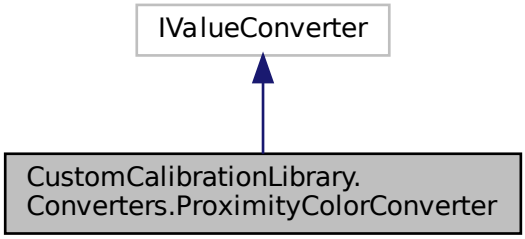
8.59 CustomCalibrationLibrary.Converters.ProximityColorConverter Class Reference

Converter class to convert the proximito of a normailezed coordinate to the center point (0.5) into colors.

Inheritance diagram for CustomCalibrationLibrary.Converters.ProximityColorConverter:



Collaboration diagram for CustomCalibrationLibrary.Converters.ProximityColorConverter:



Public Member Functions

- object Convert (object value, Type targetType, object parameter, CultureInfo culture)

  Value converter.
- object ConvertBack (object value, Type targetType, object parameter, CultureInfo culture)

  Reverted value converter.

8.59.1 Detailed Description

Converter class to convert the proximito of a normailezed coordinate to the center point (0.5) into colors.

8.59.2 Member Function Documentation

8.59.2.1 Convert()

```
object CustomCalibrationLibrary.Converters.ProximityColorConverter.Convert (
            object value,
            Type targetType,
            object parameter,
            CultureInfo culture )  [inline]
```

Value converter.

Parameters

value	The value to convert.
targetType	The type of the target value.
parameter	The conversion parameter.
culture	The language localisation.

Returns

The converted value object

8.59.2.2 ConvertBack()

```
object CustomCalibrationLibrary.Converters.ProximityColorConverter.ConvertBack (
            object value,
            Type targetType,
            object parameter,
            CultureInfo culture )  [inline]
```

Reverted value converter.

Parameters

value	The value to convert.
targetType	The type of the target value.
parameter	The conversion parameter.
culture	The language localisation.

Returns

The converted value object

Exceptions

NotSupportedException	

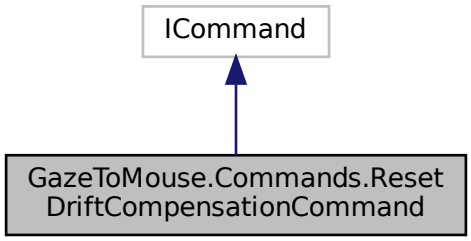
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Converters/ProximityColorConverter.cs

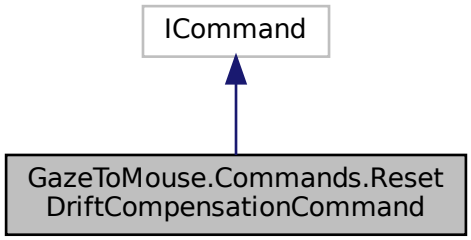
8.60 GazeToMouse.Commands.ResetDriftCompensationCommand Class Reference

Command class to reset the drift compensation.

Inheritance diagram for GazeToMouse.Commands.ResetDriftCompensationCommand:



Collaboration diagram for GazeToMouse.Commands.ResetDriftCompensationCommand:



Public Member Functions

- ResetDriftCompensationCommand (App app)

  Initializes a new instance of the ResetDriftCompensationCommand class.
- bool CanExecute (object? parameter)

  Returns whether command can be executed or not.
- void Execute (object? parameter)

  Reset the drift compensation.

Properties

- EventHandler? CanExecuteChanged

  Event handler on can executed flag change.

8.60.1 Detailed Description

Command class to reset the drift compensation.

8.60.2 Constructor & Destructor Documentation

8.60.2.1 ResetDriftCompensationCommand()

```
GazeToMouse.Commands.ResetDriftCompensationCommand.ResetDriftCompensationCommand (
            App app )  [inline]
```

Initializes a new instance of the ResetDriftCompensationCommand class.

Parameters

app	The main application

8.60.3 Member Function Documentation

8.60.3.1 CanExecute()

```
bool GazeToMouse.Commands.ResetDriftCompensationCommand.CanExecute (
            object?  parameter )  [inline]
```

Returns whether command can be executed or not.

Parameters

parameter	The command parameter

Returns

True

8.60.3.2 Execute()

```
void GazeToMouse.Commands.ResetDriftCompensationCommand.Execute (
            object?  parameter )  [inline]
```

Reset the drift compensation.

Parameters

parameter	The command parameter

8.60.4 Property Documentation

8.60.4.1 CanExecuteChanged

```
EventHandler?  GazeToMouse.Commands.ResetDriftCompensationCommand.CanExecuteChanged  [add],
[remove]
```

Event handler on can executed flag change.

The documentation for this class was generated from the following file:

- source/GazeToMouse/Commands/ResetDriftCompensationCommand.cs

8.61 GazeUtilityLibrary.ScreenArea Class Reference

The class describing the Screen area in 3d and 2d space.

Public Member Functions

- ScreenArea (Vector3 bottomLeft, Vector3 bottomRight, Vector3 topLeft, Vector3 topRight, float width, float height)

  Constructor. Assigns parameters ann computes the transformation matrix to transform a 3d point into a 2d point.
- Vector3? GetIntersectionPoint (Vector3 gazeOrigin, Vector3 gazeDirection)

  Compute the intersection point with the screen plane given a gaze origin and a gaze direction. Note that this does not compute the intersection with the screen area but with the infinite plane which is co-aligned with the screen. Pass the here computed intersection point to the method GetPoint2dNormalized to get the normalized intersection point on the sreen area.
- Vector2 GetPoint2d (Vector3 point)

  Get the 2d point on the sreen plane given a 3d point on the screen plane.
- Vector2 GetPoint2dNormalized (Vector3 point3d)

  Get the normalized 2d point on the sreen plane given a 3d point on the screen plane. Note that values outside of the interval [0, 1] indicate an intersection point outsate of the screen area.
- Vector3 GetPoint3d (Vector2 point2d)

  Get the 3d point on the sreen plane given a 2d point on the screen.
- bool Dump (string path, string prefix)

  Dump the four screen corner points to a csv file

Properties

- float Width `[get]`

  The width of the screen.
- float Height `[get]`

  The height of the screen.
- Vector3 BottomLeft `[get]`

  The coordinates of the bottom left point of the screen.
- Vector3 BottomRight `[get]`

  The coordinates of the bottom right point of the screen.
- Vector3 TopLeft `[get]`

  The coordinates of the top left point of the screen.
- Vector3 TopRight `[get]`

  The coordinates of the to right point of the screen.
- Vector3 Center `[get]`

  The coordinates of the center point of the screen.

8.61.1 Detailed Description

The class describing the Screen area in 3d and 2d space.

8.61.2 Constructor & Destructor Documentation

8.61.2.1 ScreenArea()

```
GazeUtilityLibrary.ScreenArea.ScreenArea (
            Vector3 bottomLeft,
            Vector3 bottomRight,
            Vector3 topLeft,
            Vector3 topRight,
            float width,
            float height )  [inline]
```

Constructor. Assigns parameters ann computes the transformation matrix to transform a 3d point into a 2d point.

Parameters

bottomLeft	The bottom left 3d coordinate of the screen.
bottomRight	The bottom right 3d coordinate of the screen.
topLeft	The top left 3d coordinate of the screen.
topRight	The top right 3d coordinate of the screen
width	The width of the screen
height	The heigth of the screen

8.61.3 Member Function Documentation

8.61.3.1 Dump()

```
bool GazeUtilityLibrary.ScreenArea.Dump (
            string path,
            string prefix )  [inline]
```

Dump the four screen corner points to a csv file

Parameters

path	The folder to store the file.
prefix	The file prefix.

Returns

8.61.3.2 GetIntersectionPoint()

```
Vector3?  GazeUtilityLibrary.ScreenArea.GetIntersectionPoint (
            Vector3 gazeOrigin,
            Vector3 gazeDirection )  [inline]
```

Compute the intersection point with the screen plane given a gaze origin and a gaze direction. Note that this does not compute the intersection with the screen area but with the infinite plane which is co-aligned with the screen. Pass the here computed intersection point to the method GetPoint2dNormalized to get the normalized intersection point on the sreen area.

Parameters

gazeOrigin	The origin of the gaze.
gazeDirection	The direction of the gaze.

Returns

The intersection point with the screen or null if no intersection point exists.

8.61.3.3 GetPoint2d()

```
Vector2 GazeUtilityLibrary.ScreenArea.GetPoint2d (
            Vector3 point )  [inline]
```

Get the 2d point on the sreen plane given a 3d point on the screen plane.

Parameters

point	The 3d point on the screen plane to convert.

Returns

The 2d point on the screen

8.61.3.4 GetPoint2dNormalized()

```
Vector2 GazeUtilityLibrary.ScreenArea.GetPoint2dNormalized (
            Vector3 point3d )  [inline]
```

Get the normalized 2d point on the sreen plane given a 3d point on the screen plane. Note that values outside of the interval [0, 1] indicate an intersection point outsate of the screen area.

Parameters

point3d	The 3d point on the screen plane to convert.

Returns

The normalized 2d point on the screen

8.61.3.5 GetPoint3d()

```
Vector3 GazeUtilityLibrary.ScreenArea.GetPoint3d (
            Vector2 point2d )  [inline]
```

Get the 3d point on the sreen plane given a 2d point on the screen.

Parameters

point2d	A normalized 2d point on the screen to convert.

Returns

The 3d point on the screen plane

8.61.4 Property Documentation

8.61.4.1 BottomLeft

```
Vector3 GazeUtilityLibrary.ScreenArea.BottomLeft  [get]
```

The coordinates of the bottom left point of the screen.

8.61.4.2 BottomRight

```
Vector3 GazeUtilityLibrary.ScreenArea.BottomRight  [get]
```

The coordinates of the bottom right point of the screen.

8.61.4.3 Center

```
Vector3 GazeUtilityLibrary.ScreenArea.Center  [get]
```

The coordinates of the center point of the screen.

8.61.4.4 Height

```
float GazeUtilityLibrary.ScreenArea.Height  [get]
```

The height of the screen.

8.61.4.5 TopLeft

```
Vector3 GazeUtilityLibrary.ScreenArea.TopLeft  [get]
```

The coordinates of the top left point of the screen.

8.61.4.6 TopRight

```
Vector3 GazeUtilityLibrary.ScreenArea.TopRight  [get]
```

The coordinates of the to right point of the screen.

8.61.4.7 Width

```
float GazeUtilityLibrary.ScreenArea.Width  [get]
```

The width of the screen.

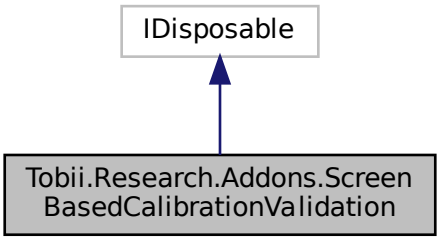
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/ScreenArea.cs

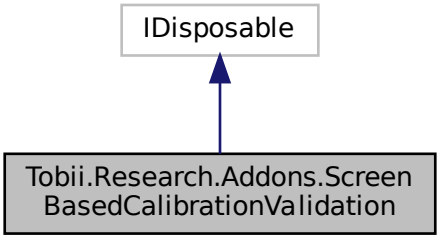
8.62 Tobii.Research.Addons.ScreenBasedCalibrationValidation Class Reference

Provides methods and properties for managing calibration validation for screen based eye trackers.

Inheritance diagram for Tobii.Research.Addons.ScreenBasedCalibrationValidation:



Collaboration diagram for Tobii.Research.Addons.ScreenBasedCalibrationValidation:



Public Types

- enum ValidationState { **NotInValidationMode**, **NotCollectingData**, **CollectingData** }

  ValidationState.NotInValidationMode - EnterValidationMode must be called starting to collect data. ValidationState.NotCollectingData - Ready to start collecting data or computing result. ValidationState.CollectingData - Currently collecting data. Will finish after the sample count is reached or a timeout.

Public Member Functions

- ScreenBasedCalibrationValidation (IEyeTracker eyeTracker, int sampleCount=30, int timeoutMS=1000)

  Create a calibration validation object for screen based eye trackers.
- void StartCollectingData (NormalizedPoint2D calibrationPointCoordinates)

  Starts collecting data for a calibration validation point.The argument used is the point the user is assumed to be looking at and is given in the active display area coordinate system. Please check State property to know when data collection is completed (or timed out).
- void DiscardData (NormalizedPoint2D calibrationPointCoordinates)

  Removes the collected data for a specific calibration validation point.
- void EnterValidationMode ()

  Enter the calibration validation mode and starts subscribing to gaze data from the eye tracker.
- void LeaveValidationMode ()

  Leaves the calibration validation mode, clears all collected data, and unsubscribes from the eye tracker.
- CalibrationValidationResult Compute ()

  Uses the collected data and tries to compute accuracy and precision values for all points. If the calculation is successful, the result is returned, and stored in the Result property of the CalibrationValidation object. If there is insufficient data to compute the results for a certain point that CalibrationValidationPoint will contain invalid data (NaN) for the results. Gaze data will still be untouched. If there is no valid data for any point, the average results of CalibrationValidationResult will be invalid (NaN) as well.
- void Dispose ()

  Dispose will unsubscribe to gaze data and exit validation mode, if the object is not already in ValidationState.NotInValidationMode
- override string ToString ()

  Convert validation values to a string.

Properties

- ValidationState State [get]

  Get the current state of the validation object.
- CalibrationValidationResult Result [get]

  Get the current CalibrationValidationResult with the computed accuracy and precision. Compute must have been called for this to contain valid data.

8.62.1 Detailed Description

Provides methods and properties for managing calibration validation for screen based eye trackers.

8.62.2 Member Enumeration Documentation

8.62.2.1 ValidationState

```
enum Tobii.Research.Addons.ScreenBasedCalibrationValidation.ValidationState [strong]
```

ValidationState.NotInValidationMode - EnterValidationMode must be called starting to collect data. ValidationState.NotCollectingData - Ready to start collecting data or computing result. ValidationState.CollectingData - Currently collecting data. Will finish after the sample count is reached or a timeout.

8.62.3 Constructor & Destructor Documentation

8.62.3.1 ScreenBasedCalibrationValidation()

```
Tobii.Research.Addons.ScreenBasedCalibrationValidation.ScreenBasedCalibrationValidation (
            IEyeTracker eyeTracker,
            int sampleCount = 30,
            int timeoutMS = 1000 ) [inline]
```

Create a calibration validation object for screen based eye trackers.

Parameters

eyeTracker	An IEyeTracker instance.
sampleCount	The number of samples to collect. Default 30, minimum 10, maximum 3000.
timeoutMS	Timeout in milliseconds. Default 1000, minimum 100, maximum 3000.

8.62.4 Member Function Documentation

8.62.4.1 Compute()

```
CalibrationValidationResult Tobii.Research.Addons.ScreenBasedCalibrationValidation.Compute ( )
[inline]
```

Uses the collected data and tries to compute accuracy and precision values for all points. If the calculation is successful, the result is returned, and stored in the Result property of the CalibrationValidation object. If there is insufficient data to compute the results for a certain point that CalibrationValidationPoint will contain invalid data (NaN) for the results. Gaze data will still be untouched. If there is no valid data for any point, the average results of CalibrationValidationResult will be invalid (NaN) as well.

Returns

The CalibrationValidationResult

8.62.4.2 DiscardData()

```
void Tobii.Research.Addons.ScreenBasedCalibrationValidation.DiscardData (
            NormalizedPoint2D calibrationPointCoordinates )  [inline]
```

Removes the collected data for a specific calibration validation point.

Parameters

calibrationPointCoordinates	The calibration point to remove.

8.62.4.3 Dispose()

```
void Tobii.Research.Addons.ScreenBasedCalibrationValidation.Dispose ( )  [inline]
```

Dispose will unsubscribe to gaze data and exit validation mode, if the object is not already in ValidationState.Not↩ InValidationMode

8.62.4.4 EnterValidationMode()

```
void Tobii.Research.Addons.ScreenBasedCalibrationValidation.EnterValidationMode ( )  [inline]
```

Enter the calibration validation mode and starts subscribing to gaze data from the eye tracker.

8.62.4.5 LeaveValidationMode()

```
void Tobii.Research.Addons.ScreenBasedCalibrationValidation.LeaveValidationMode ( )  [inline]
```

Leaves the calibration validation mode, clears all collected data, and unsubscribes from the eye tracker.

8.62.4.6 StartCollectingData()

```
void Tobii.Research.Addons.ScreenBasedCalibrationValidation.StartCollectingData (
            NormalizedPoint2D calibrationPointCoordinates )  [inline]
```

Starts collecting data for a calibration validation point.The argument used is the point the user is assumed to be looking at and is given in the active display area coordinate system. Please check State property to know when data collection is completed (or timed out).

Parameters

calibrationPointCoordinates	The normalized 2D point on the display area

8.62.4.7 ToString()

```
override string Tobii.Research.Addons.ScreenBasedCalibrationValidation.ToString ( )  [inline]
```

Convert validation values to a string.

Returns

The validation string.

8.62.5 Property Documentation

8.62.5.1 Result

```
CalibrationValidationResult Tobii.Research.Addons.ScreenBasedCalibrationValidation.Result
[get]
```

Get the current CalibrationValidationResult with the computed accuracy and precision. Compute must have been called for this to contain valid data.

8.62.5.2 State

```
ValidationState Tobii.Research.Addons.ScreenBasedCalibrationValidation.State  [get]
```

Get the current state of the validation object.

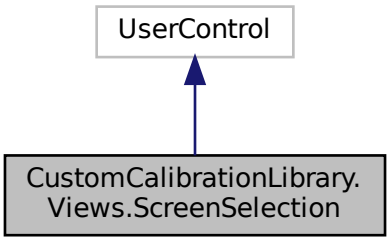
The documentation for this class was generated from the following file:

- source/TobiiProSdkAddons/ScreenBasedCalibrationValidation.cs

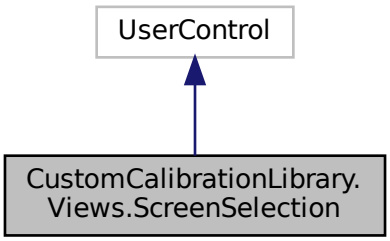
8.63 CustomCalibrationLibrary.Views.ScreenSelection Class Reference

Interaction logic for ScreenSelection.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.ScreenSelection:



Collaboration diagram for CustomCalibrationLibrary.Views.ScreenSelection:



Public Member Functions

- ScreenSelection (CalibrationModel model, Window window)

  Initializes a new instance of the ScreenSelection class.

8.63.1 Detailed Description

Interaction logic for ScreenSelection.xaml

8.63.2 Constructor & Destructor Documentation

8.63.2.1 ScreenSelection()

```
CustomCalibrationLibrary.Views.ScreenSelection.ScreenSelection (
            CalibrationModel model,
            Window window )  [inline]
```

Initializes a new instance of the ScreenSelection class.

Parameters

model	The calibration model.
window	The target window.

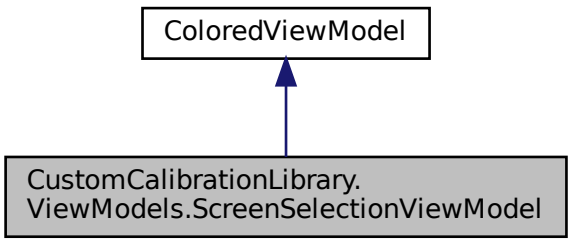
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/ScreenSelection.xaml.cs

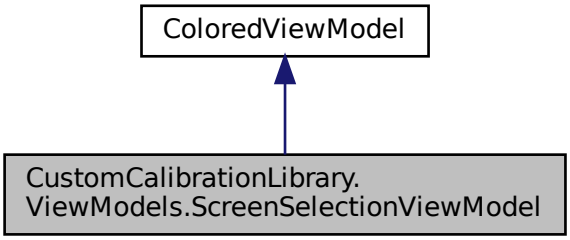
8.64 CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel Class Reference

The view model class for the screen selection view.

Inheritance diagram for CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel:



Collaboration diagram for CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel:



Public Member Functions

- ScreenSelectionViewModel (CalibrationModel model, Window window)

  Initializes a new instance of the ScreenSelectionViewModel class.

Properties

- ObservableCollection< Monitor > Monitors `[get]`

  The observable lidt of monitors to select from.
- ICommand CalibrationStartCommand `[get]`

  Command to start the calibration
- ICommand CalibrationAbortCommand `[get]`

  Command to abort the calibration
- ICommand ScreenSwitchCommand `[get]`

  Command to switch the screen

8.64.1 Detailed Description

The view model class for the screen selection view.

8.64.2 Constructor & Destructor Documentation

8.64.2.1 ScreenSelectionViewModel()

```
CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.ScreenSelectionViewModel (
            CalibrationModel model,
            Window window )  [inline]
```

Initializes a new instance of the ScreenSelectionViewModel class.

Parameters

model	The calibration model
window	The target window of the screen selection

8.64.3 Property Documentation

8.64.3.1 CalibrationAbortCommand

```
ICommand CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.CalibrationAbortCommand
[get]
```

Command to abort the calibration

8.64.3.2 CalibrationStartCommand

```
ICommand CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.CalibrationStartCommand
[get]
```

Command to start the calibration

8.64.3.3 Monitors

```
ObservableCollection<Monitor> CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.↩
Monitors  [get]
```

The observable lidt of monitors to select from.

8.64.3.4 ScreenSwitchCommand

```
ICommand CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.ScreenSwitchCommand
[get]
```

Command to switch the screen

The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/ScreenSelectionViewModel.cs

8.65 GazeUtilityLibrary.ScreenTriangle Class Reference

A class to describe a triangle. This was supposed to be used to construct the ScreenArea but it turned out that it is simpler to work with the screen plane and use the normalised intersection points to check wheter the gaze point is outside the screen area.

Public Member Functions

- ScreenTriangle (Vector3 v1, Vector3 v2, Vector3 v3)

  Initializes a new instance of the ScreenTriangle class.
- Vector3? GetIntersectionPoint (Vector3 origin, Vector3 direction)

  Compute the intersection point with the triangle with the Moller-Trumbore algorithm.

Properties

- Vector3 V1 `[get]`

  A corner point of the triangle.
- Vector3 V2 `[get]`

  A corner point of the triangle.
- Vector3 V3 `[get]`

  A corner point of the triangle.
- Vector3 E1 `[get]`

  The edge vector from v1 to v2.
- Vector3 E2 `[get]`

  The edge vector from v1 to v3.

8.65.1 Detailed Description

A class to describe a triangle. This was supposed to be used to construct the ScreenArea but it turned out that it is simpler to work with the screen plane and use the normalised intersection points to check wheter the gaze point is outside the screen area.

8.65.2 Constructor & Destructor Documentation

8.65.2.1 ScreenTriangle()

```
GazeUtilityLibrary.ScreenTriangle.ScreenTriangle (
            Vector3 v1,
            Vector3 v2,
            Vector3 v3 )  [inline]
```

Initializes a new instance of the ScreenTriangle class.

Parameters

v1	A corner point of the triangle.
v2	A corner point of the triangle.
v3	A corner point of the triangle.

8.65.3 Member Function Documentation

8.65.3.1 GetIntersectionPoint()

```
Vector3?  GazeUtilityLibrary.ScreenTriangle.GetIntersectionPoint (
            Vector3 origin,
            Vector3 direction )  [inline]
```

Compute the intersection point with the triangle with the Moller-Trumbore algorithm.

Parameters

origin	The origin of the gaze point
direction	The direction of the gaze point

Returns

The intersection point or null if no intersection point could be computed.

8.65.4 Property Documentation

8.65.4.1 E1

```
Vector3 GazeUtilityLibrary.ScreenTriangle.E1  [get]
```

The edge vector from v1 to v2.

8.65.4.2 E2

```
Vector3 GazeUtilityLibrary.ScreenTriangle.E2  [get]
```

The edge vector from v1 to v3.

8.65.4.3 V1

```
Vector3 GazeUtilityLibrary.ScreenTriangle.V1  [get]
```

A corner point of the triangle.

8.65.4.4 V2

```
Vector3 GazeUtilityLibrary.ScreenTriangle.V2  [get]
```

A corner point of the triangle.

8.65.4.5 V3

```
Vector3 GazeUtilityLibrary.ScreenTriangle.V3  [get]
```

A corner point of the triangle.

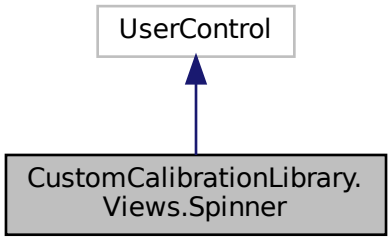
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/ScreenTriangle.cs

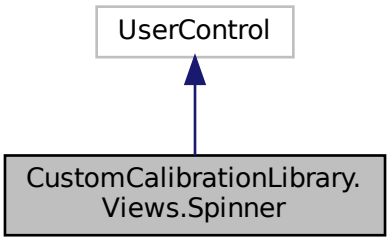
8.66 CustomCalibrationLibrary.Views.Spinner Class Reference

Interaction logic for Computing.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.Spinner:



Collaboration diagram for CustomCalibrationLibrary.Views.Spinner:



Public Member Functions

- Spinner (Color backgroundColor)

  Initializes a new instance of the Computing class.

8.66.1 Detailed Description

Interaction logic for Computing.xaml

8.66.2 Constructor & Destructor Documentation

8.66.2.1 Spinner()

```
CustomCalibrationLibrary.Views.Spinner.Spinner (
            Color backgroundColor )  [inline]
```

Initializes a new instance of the Computing class.

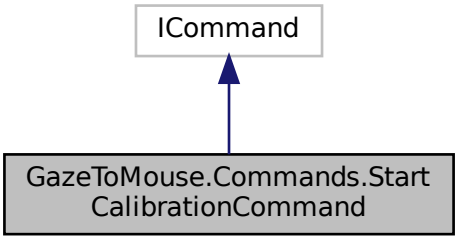
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/Spinner.xaml.cs

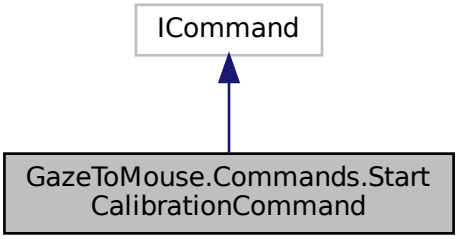
8.67 GazeToMouse.Commands.StartCalibrationCommand Class Reference

Command class start the calibration.

Inheritance diagram for GazeToMouse.Commands.StartCalibrationCommand:



Collaboration diagram for GazeToMouse.Commands.StartCalibrationCommand:



Public Member Functions

- StartCalibrationCommand (App app)

  Initializes a new instance of the StartCalibrationCommand class.
- bool CanExecute (object? parameter)

  Returns whether command can be executed or not.
- void Execute (object? parameter)

  Start the calibration.

Properties

- EventHandler? CanExecuteChanged

  Event handler on can executed flag change.

8.67.1 Detailed Description

Command class start the calibration.

8.67.2 Constructor & Destructor Documentation

8.67.2.1 StartCalibrationCommand()

```
GazeToMouse.Commands.StartCalibrationCommand.StartCalibrationCommand (
            App app )  [inline]
```

Initializes a new instance of the StartCalibrationCommand class.

Parameters

app	The main application

8.67.3 Member Function Documentation

8.67.3.1 CanExecute()

```
bool GazeToMouse.Commands.StartCalibrationCommand.CanExecute (
            object?  parameter )  [inline]
```

Returns whether command can be executed or not.

Parameters

parameter	The command parameter

Returns

True

8.67.3.2 Execute()

```
void GazeToMouse.Commands.StartCalibrationCommand.Execute (
            object?  parameter )  [inline]
```

Start the calibration.

Parameters

parameter	The command parameter

8.67.4 Property Documentation

8.67.4.1 CanExecuteChanged

```
EventHandler? GazeToMouse.Commands.StartCalibrationCommand.CanExecuteChanged [add], [remove]
```

Event handler on can executed flag change.

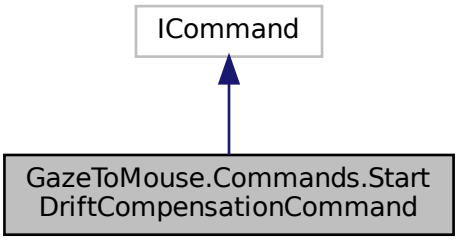
The documentation for this class was generated from the following file:

- source/GazeToMouse/Commands/StartCalibrationCommand.cs

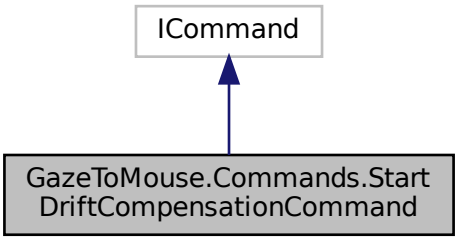
8.68 GazeToMouse.Commands.StartDriftCompensationCommand Class Reference

Command class to start the drift compensation.

Inheritance diagram for GazeToMouse.Commands.StartDriftCompensationCommand:



Collaboration diagram for GazeToMouse.Commands.StartDriftCompensationCommand:



Public Member Functions

- StartDriftCompensationCommand (App app)

  Initializes a new instance of the StartDriftCompensationCommand class.
- bool CanExecute (object? parameter)

  Returns whether command can be executed or not.
- void Execute (object? parameter)

  Start the drift compensation.

Properties

- EventHandler? CanExecuteChanged

  Event handler on can executed flag change.

8.68.1 Detailed Description

Command class to start the drift compensation.

8.68.2 Constructor & Destructor Documentation

8.68.2.1 StartDriftCompensationCommand()

```
GazeToMouse.Commands.StartDriftCompensationCommand.StartDriftCompensationCommand (
            App app )  [inline]
```

Initializes a new instance of the StartDriftCompensationCommand class.

Parameters

app	The main application

8.68.3 Member Function Documentation

8.68.3.1 CanExecute()

```
bool GazeToMouse.Commands.StartDriftCompensationCommand.CanExecute (
            object?  parameter )  [inline]
```

Returns whether command can be executed or not.

Parameters

parameter	The command parameter

Returns

True

8.68.3.2 Execute()

```
void GazeToMouse.Commands.StartDriftCompensationCommand.Execute (
            object?  parameter )  [inline]
```

Start the drift compensation.

Parameters

parameter	The command parameter

8.68.4 Property Documentation

8.68.4.1 CanExecuteChanged

```
EventHandler?  GazeToMouse.Commands.StartDriftCompensationCommand.CanExecuteChanged  [add],
[remove]
```

Event handler on can executed flag change.

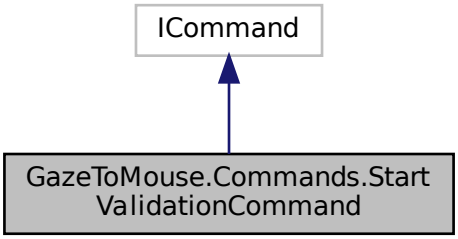
The documentation for this class was generated from the following file:

- source/GazeToMouse/Commands/StartDriftCompensationCommand.cs

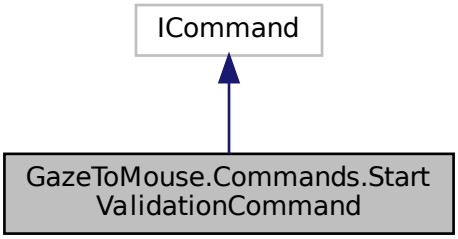
8.69 GazeToMouse.Commands.StartValidationCommand Class Reference

Command class to start the validation.

Inheritance diagram for GazeToMouse.Commands.StartValidationCommand:



Collaboration diagram for GazeToMouse.Commands.StartValidationCommand:



Public Member Functions

- StartValidationCommand (App app)

  Initializes a new instance of the StartValidationCommand class.
- bool CanExecute (object? parameter)

  Returns whether command can be executed or not.
- void Execute (object? parameter)

  Start the validation.

Properties

- EventHandler? CanExecuteChanged

  Event handler on can executed flag change.

8.69.1 Detailed Description

Command class to start the validation.

8.69.2 Constructor & Destructor Documentation

8.69.2.1 StartValidationCommand()

```
GazeToMouse.Commands.StartValidationCommand.StartValidationCommand (
            App app ) [inline]
```

Initializes a new instance of the StartValidationCommand class.

Parameters

app	The main application

8.69.3 Member Function Documentation

8.69.3.1 CanExecute()

```
bool GazeToMouse.Commands.StartValidationCommand.CanExecute (
            object? parameter ) [inline]
```

Returns whether command can be executed or not.

Parameters

parameter	The command parameter

Returns

True

8.69.3.2 Execute()

```
void GazeToMouse.Commands.StartValidationCommand.Execute (
            object?  parameter )  [inline]
```

Start the validation.

Parameters

parameter	The command parameter

8.69.4 Property Documentation

8.69.4.1 CanExecuteChanged

```
EventHandler?  GazeToMouse.Commands.StartValidationCommand.CanExecuteChanged  [add], [remove]
```

Event handler on can executed flag change.

The documentation for this class was generated from the following file:

- source/GazeToMouse/Commands/StartValidationCommand.cs

8.70 GazeUtilityLibrary.TrackerLogger Class Reference

Simple logger class.

Public Member Functions

- TrackerLogger (string? logPath, EOutputType type=EOutputType.gaze)

  Initializes a new instance of the TrackerLogger class.
- void DumpFatal (Exception e)

  Dumps exception to a new file if it is not possible to write to the main log file.
- void Debug (string message)

  wrapper function for debug level logging.
- void Info (string message)

  wrapper function for info level logging
- void Warning (string message)

  wrapper function for warning level logging
- void Error (string message)

  wrapper function for error level logging

8.70.1 Detailed Description

Simple logger class.

8.70.2 Constructor & Destructor Documentation

8.70.2.1 TrackerLogger()

```
GazeUtilityLibrary.TrackerLogger.TrackerLogger (
            string?  logPath,
            EOutputType type = EOutputType.gaze )  [inline]
```

Initializes a new instance of the TrackerLogger class.

8.70.3 Member Function Documentation

8.70.3.1 Debug()

```
void GazeUtilityLibrary.TrackerLogger.Debug (
            string message )  [inline]
```

wrapper function for debug level logging.

Parameters

message	The message.

8.70.3.2 DumpFatal()

```
void GazeUtilityLibrary.TrackerLogger.DumpFatal (
            Exception e )  [inline]
```

Dumps exception to a new file if it is not possible to write to the main log file.

Parameters

e	The exception.

8.70.3.3 Error()

```
void GazeUtilityLibrary.TrackerLogger.Error (
            string message )  [inline]
```

wrapper function for error level logging

Parameters

message	The message.

8.70.3.4 Info()

```
void GazeUtilityLibrary.TrackerLogger.Info (
            string message )  [inline]
```

wrapper function for info level logging

Parameters

message	The message.

8.70.3.5 Warning()

```
void GazeUtilityLibrary.TrackerLogger.Warning (
            string message )  [inline]
```

wrapper function for warning level logging

Parameters

message	The message.

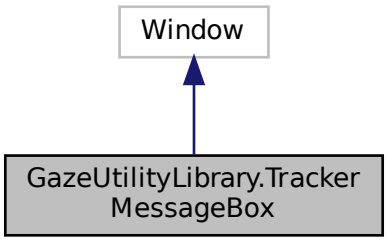
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Logger.cs

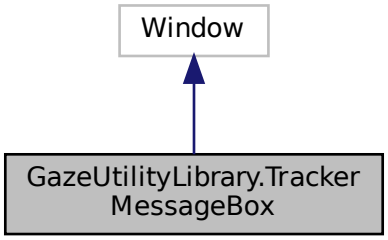
8.71 GazeUtilityLibrary.TrackerMessageBox Class Reference

Interaction logic for TrackerMessageBox.xaml

Inheritance diagram for GazeUtilityLibrary.TrackerMessageBox:



Collaboration diagram for GazeUtilityLibrary.TrackerMessageBox:



8.71.1 Detailed Description

Interaction logic for TrackerMessageBox.xaml

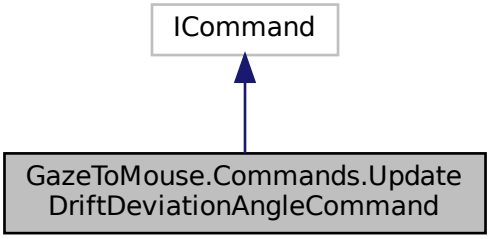
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/TrackerMessageBox.xaml.cs

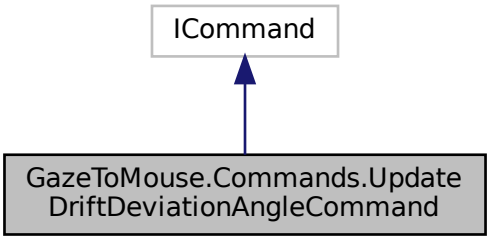
8.72 GazeToMouse.Commands.UpdateDriftDeviationAngleCommand Class Reference

Command class to start the drift compensation.

Inheritance diagram for GazeToMouse.Commands.UpdateDriftDeviationAngleCommand:



Collaboration diagram for GazeToMouse.Commands.UpdateDriftDeviationAngleCommand:



Public Member Functions

- UpdateDriftDeviationAngleCommand (App app, Func< double, double > lambda)

  Initializes a new instance of the UpdateDriftDeviationAngleCommand class.
- bool CanExecute (object? parameter)

  Returns whether command can be executed or not.
- void Execute (object? parameter)

  Start the drift compensation.

Properties

- EventHandler? CanExecuteChanged

  Event handler on can executed flag change.

8.72.1 Detailed Description

Command class to start the drift compensation.

8.72.2 Constructor & Destructor Documentation

8.72.2.1 UpdateDriftDeviationAngleCommand()

```
GazeToMouse.Commands.UpdateDriftDeviationAngleCommand.UpdateDriftDeviationAngleCommand (
            App app,
            Func< double, double > lambda )  [inline]
```

Initializes a new instance of the UpdateDriftDeviationAngleCommand class.

Parameters

app	The main application
lambda	A delegate to set the drift deviation angle.

8.72.3 Member Function Documentation

8.72.3.1 CanExecute()

```
bool GazeToMouse.Commands.UpdateDriftDeviationAngleCommand.CanExecute (
            object?  parameter )  [inline]
```

Returns whether command can be executed or not.

Parameters

parameter	The command parameter

Returns

True

8.72.3.2 Execute()

```
void GazeToMouse.Commands.UpdateDriftDeviationAngleCommand.Execute (
            object?  parameter )  [inline]
```

Start the drift compensation.

Parameters

parameter	The command parameter

8.72.4 Property Documentation

8.72.4.1 CanExecuteChanged

```
EventHandler? GazeToMouse.Commands.UpdateDriftDeviationAngleCommand.CanExecuteChanged [add],
[remove]
```

Event handler on can executed flag change.

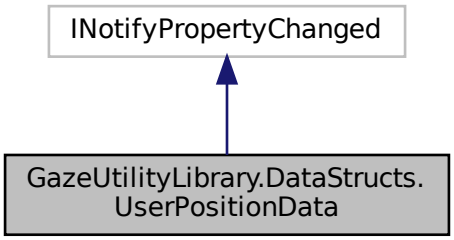
The documentation for this class was generated from the following file:

- source/GazeToMouse/Commands/UpdateDriftDeviationAngleCommand.cs

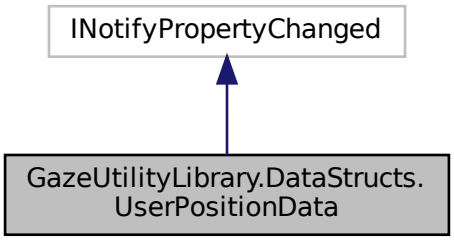
8.73 GazeUtilityLibrary.DataStructs.UserPositionData Class Reference

The user position to be rendered on the screen.

Inheritance diagram for GazeUtilityLibrary.DataStructs.UserPositionData:



Collaboration diagram for GazeUtilityLibrary.DataStructs.UserPositionData:



Public Member Functions

- UserPositionData ()

  Initializes a new instance of the UserPositionData class.
- UserPositionData (double xCoordLeft, double yCoordLeft, double zCoordLeft, double xCoordRight, double yCoordRight, double zCoordRight)

  Initializes a new instance of the UserPositionData class.

Properties

- double XCoordLeft `[get, set]`

  The normalized x coordinate of the left eye.
- double YCoordLeft `[get, set]`

  The normalized y coordinate of the left eye.
- double ZCoordLeft `[get, set]`

  The normalized z coordinate of the left eye.
- double XCoordRight `[get, set]`

  The normalized x coordinate of the right eye.
- double YCoordRight `[get, set]`

  The normalized y coordinate of the right eye.
- double ZCoordRight `[get, set]`

  The normalized z coordinate of the right eye.

Events

- PropertyChangedEventHandler? PropertyChanged

  The property change event handler.

8.73.1 Detailed Description

The user position to be rendered on the screen.

8.73.2 Constructor & Destructor Documentation

8.73.2.1 UserPositionData() [1/2]

```
GazeUtilityLibrary.DataStructs.UserPositionData.UserPositionData ( )  [inline]
```

Initializes a new instance of the UserPositionData class.

8.73.2.2 UserPositionData() [2/2]

```
GazeUtilityLibrary.DataStructs.UserPositionData.UserPositionData (
            double xCoordLeft,
            double yCoordLeft,
            double zCoordLeft,
            double xCoordRight,
            double yCoordRight,
            double zCoordRight )  [inline]
```

Initializes a new instance of the UserPositionData class.

Parameters

xCoordLeft	The normalized x coordinate of the left eye.
yCoordLeft	The normalized y coordinate of the left eye.
zCoordLeft	The normalized z coordinate of the left eye.
xCoordRight	The normalized x coordinate of the right eye.
yCoordRight	The normalized y coordinate of the right eye.
zCoordRight	The normalized z coordinate of the right eye.

8.73.3 Property Documentation

8.73.3.1 XCoordLeft

```
double GazeUtilityLibrary.DataStructs.UserPositionData.XCoordLeft  [get], [set]
```

The normalized x coordinate of the left eye.

8.73.3.2 XCoordRight

```
double GazeUtilityLibrary.DataStructs.UserPositionData.XCoordRight  [get], [set]
```

The normalized x coordinate of the right eye.

8.73.3.3 YCoordLeft

```
double GazeUtilityLibrary.DataStructs.UserPositionData.YCoordLeft  [get], [set]
```

The normalized y coordinate of the left eye.

8.73.3.4 YCoordRight

```
double GazeUtilityLibrary.DataStructs.UserPositionData.YCoordRight  [get], [set]
```

The normalized y coordinate of the right eye.

8.73.3.5 ZCoordLeft

```
double GazeUtilityLibrary.DataStructs.UserPositionData.ZCoordLeft  [get], [set]
```

The normalized z coordinate of the left eye.

8.73.3.6 ZCoordRight

```
double GazeUtilityLibrary.DataStructs.UserPositionData.ZCoordRight  [get], [set]
```

The normalized z coordinate of the right eye.

8.73.4 Event Documentation

8.73.4.1 PropertyChanged

```
PropertyChangedEventHandler?  GazeUtilityLibrary.DataStructs.UserPositionData.PropertyChanged
```

The property change event handler.

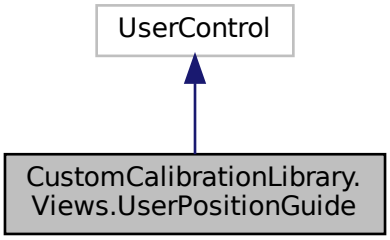
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/UserPositionData.cs

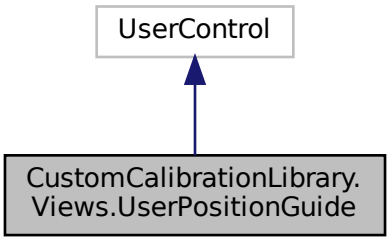
8.74 CustomCalibrationLibrary.Views.UserPositionGuide Class Reference

Interaction logic for UserPositionGuide.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.UserPositionGuide:



Collaboration diagram for CustomCalibrationLibrary.Views.UserPositionGuide:



Public Member Functions

- UserPositionGuide (CalibrationModel model)

  Initializes a new instance of the UserPositionGuide class.

8.74.1 Detailed Description

Interaction logic for UserPositionGuide.xaml

8.74.2 Constructor & Destructor Documentation

8.74.2.1 UserPositionGuide()

```
CustomCalibrationLibrary.Views.UserPositionGuide.UserPositionGuide (
            CalibrationModel model ) [inline]
```

Initializes a new instance of the UserPositionGuide class.

Parameters

model	The calibration model.

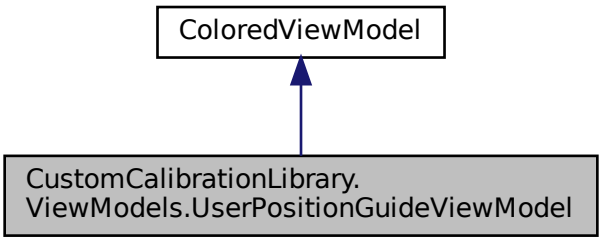
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/UserPositionGuide.xaml.cs

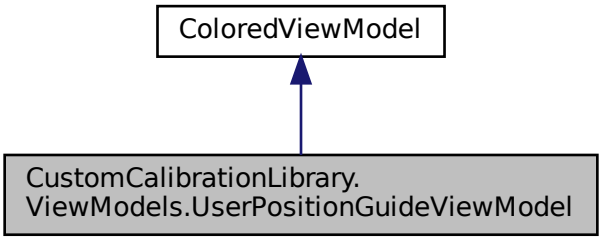
8.75 CustomCalibrationLibrary.ViewModels.UserPositionGuideView↩ Model Class Reference

The view model class for the user position guide view.

Inheritance diagram for CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel:



Collaboration diagram for CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel:



Public Member Functions

- UserPositionGuideViewModel (CalibrationModel model)
  Constructor

Properties

- UserPositionData UserPosition `[get]`
  The user position to be represented on the view
- ICommand CalibrationStartCommand `[get]`
  Command to start the calibration
- ICommand CalibrationAbortCommand `[get]`
  Command to abort the calibration

8.75.1 Detailed Description

The view model class for the user position guide view.

8.75.2 Constructor & Destructor Documentation

8.75.2.1 UserPositionGuideViewModel()

```
CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel.UserPositionGuideViewModel (
            CalibrationModel model ) [inline]
```

Constructor

Parameters

model	The calibartion model

8.75.3 Property Documentation

8.75.3.1 CalibrationAbortCommand

```
ICommand CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel.CalibrationAbort↩
Command [get]
```

Command to abort the calibration

8.75.3.2 CalibrationStartCommand

```
ICommand CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel.CalibrationStart↩
Command  [get]
```

Command to start the calibration

8.75.3.3 UserPosition

```
UserPositionData CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel.UserPosition
[get]
```

The user position to be represented on the view

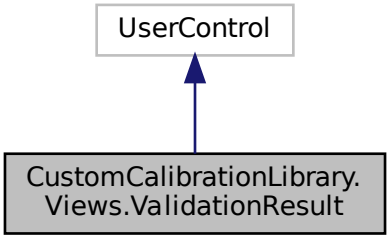
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/UserPositionGuideViewModel.cs

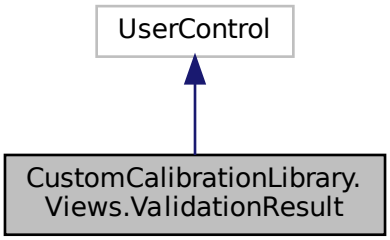
8.76 CustomCalibrationLibrary.Views.ValidationResult Class Reference

Interaction logic for ValidationResult.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.ValidationResult:



Collaboration diagram for CustomCalibrationLibrary.Views.ValidationResult:



Public Member Functions

- ValidationResult (CalibrationModel model)

  Initializes a new instance of the ValidationResult class.

8.76.1 Detailed Description

Interaction logic for ValidationResult.xaml

8.76.2 Constructor & Destructor Documentation

8.76.2.1 ValidationResult()

```
CustomCalibrationLibrary.Views.ValidationResult.ValidationResult (
            CalibrationModel model )  [inline]
```

Initializes a new instance of the ValidationResult class.

Parameters

model	The calibration model.

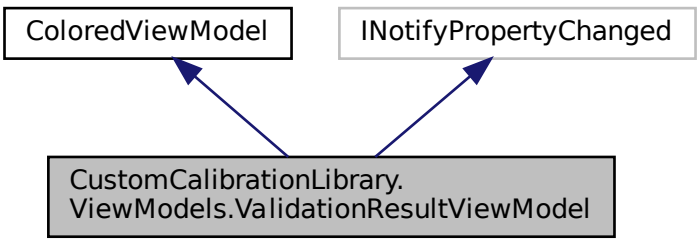
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Views/ValidationResult.xaml.cs

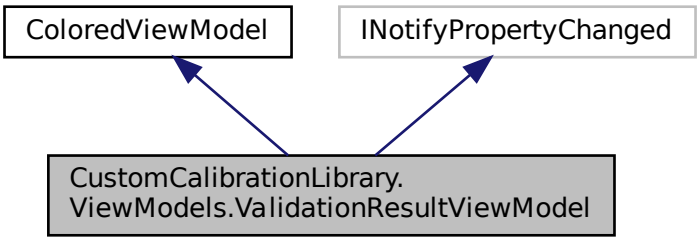
8.77 CustomCalibrationLibrary.ViewModels.ValidationResultViewModel Class Reference

View model class of the gaze validation result.

Inheritance diagram for CustomCalibrationLibrary.ViewModels.ValidationResultViewModel:



Collaboration diagram for CustomCalibrationLibrary.ViewModels.ValidationResultViewModel:



Public Member Functions

- ValidationResultViewModel (CalibrationModel model)

  Constructor

Properties

- ICommand ValidationRestartCommand `[get]`

  Command to restart the validation
- ICommand ValidationCloseCommand `[get]`

  Command to close the validation window
- GazeValidationData ValidationData `[get]`

  The validation result
- Visibility SuccessVisibility `[get]`

  The visibility flag for all items if the accuracy is acceptable.
- Visibility AlertVisibility `[get]`

  The visibility flag for all items if the accuracy is too low.
- Visibility RedoTimerVisibility `[get]`

  The visibility flag for all items if the accuracy is too low.
- int RemainingSec `[get, set]`

  The number or remaining seconds before an automatic calibration restart.

Events

- PropertyChangedEventHandler? PropertyChanged

  The protperty changed handler.

8.77.1 Detailed Description

View model class of the gaze validation result.

8.77.2 Constructor & Destructor Documentation

8.77.2.1 ValidationResultViewModel()

```
CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.ValidationResultViewModel (
            CalibrationModel model )  [inline]
```

Constructor

Parameters

model	The claibration model

8.77.3 Property Documentation

8.77.3.1 AlertVisibility

```
Visibility CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.AlertVisibility  [get]
```

The visibility flag for all items if the accuracy is too low.

8.77.3.2 RedoTimerVisibility

```
Visibility CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.RedoTimerVisibility
[get]
```

The visibility flag for all items if the accuracy is too low.

8.77.3.3 RemainingSec

```
int CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.RemainingSec  [get], [set]
```

The number or remaining seconds before an automatic calibration restart.

8.77.3.4 SuccessVisibility

```
Visibility CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.SuccessVisibility
[get]
```

The visibility flag for all items if the accuracy is acceptable.

8.77.3.5 ValidationCloseCommand

```
ICommand CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.ValidationCloseCommand
[get]
```

Command to close the validation window

8.77.3.6 ValidationData

```
GazeValidationData CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.Validation↩
Data [get]
```

The validation result

8.77.3.7 ValidationRestartCommand

```
ICommand CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.ValidationRestart↩
Command [get]
```

Command to restart the validation

8.77.4 Event Documentation

8.77.4.1 PropertyChanged

```
PropertyChangedEventHandler? CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.↩
PropertyChanged
```

The protperty changed handler.

The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/ValidationResultViewModel.cs

Index