

# Gaze Toolset

v3.2.0

Generated by Doxygen 1.8.17



<b>1 v3.2.1</b>	<b>1</b>
<b>2 Toolset to Control Tobii Eye Tracker</b>	<b>7</b>
<b>3 Namespace Index</b>	<b>11</b>
3.1 Namespace List . . . . .	11
<b>4 Hierarchical Index</b>	<b>13</b>
4.1 Class Hierarchy . . . . .	13
<b>5 Class Index</b>	<b>15</b>
5.1 Class List . . . . .	15
<b>6 Namespace Documentation</b>	<b>19</b>
6.1 CustomCalibrationLibrary Namespace Reference . . . . .	19
6.2 CustomCalibrationLibrary.Commands Namespace Reference . . . . .	19
6.3 CustomCalibrationLibrary.Converters Namespace Reference . . . . .	19
6.4 CustomCalibrationLibrary.Models Namespace Reference . . . . .	19
6.4.1 Enumeration Type Documentation . . . . .	20
6.4.1.1 CalibrationEventType . . . . .	20
6.4.1.2 CalibrationStatus . . . . .	20
6.5 CustomCalibrationLibrary.ViewModels Namespace Reference . . . . .	20
6.6 CustomCalibrationLibrary.Views Namespace Reference . . . . .	21
6.7 GazeControl Namespace Reference . . . . .	21
6.8 GazeToMouse Namespace Reference . . . . .	21
6.9 GazeUtilityLibrary Namespace Reference . . . . .	22
6.9.1 Detailed Description . . . . .	22
6.9.2 Enumeration Type Documentation . . . . .	23
6.9.2.1 ECalibrationDataError . . . . .	23
6.9.2.2 EGazeConfigError . . . . .	23
6.9.2.3 EGazeDataError . . . . .	23
6.10 GazeUtilityLibrary.DataStructs Namespace Reference . . . . .	23
6.10.1 Enumeration Type Documentation . . . . .	24
6.10.1.1 CalibrationOutputValue . . . . .	24
6.10.1.2 GazeOutputValue . . . . .	24
6.10.1.3 ValidationOutputValue . . . . .	25
6.11 GazeUtilityLibrary.Tracker Namespace Reference . . . . .	25
6.12 ShowMouse Namespace Reference . . . . .	25
6.13 Tobii Namespace Reference . . . . .	25
6.14 Tobii.Research Namespace Reference . . . . .	25
6.15 Tobii.Research.Addons Namespace Reference . . . . .	25
6.16 Tobii.Research.Addons.Utility Namespace Reference . . . . .	26
6.17 TobiiCalibrate Namespace Reference . . . . .	26

<b>7 Class Documentation</b>	<b>27</b>
7.1 GazeControl.App Class Reference	27
7.1.1 Detailed Description	28
7.2 ShowMouse.App Class Reference	28
7.2.1 Detailed Description	28
7.3 TobiiCalibrate.App Class Reference	29
7.3.1 Detailed Description	29
7.4 GazeToMouse.App Class Reference	30
7.4.1 Detailed Description	31
7.4.2 Constructor & Destructor Documentation	31
7.4.2.1 App()	31
7.4.3 Member Function Documentation	31
7.4.3.1 CalibrationValidate()	31
7.4.3.2 CompensateDrift()	32
7.4.3.3 CustomCalibrate()	32
7.4.3.4 GazeRecordingDisable()	32
7.4.3.5 GazeRecordingEnable()	32
7.4.3.6 MouseTrackingDisable()	32
7.4.3.7 MouseTrackingEnable()	33
7.4.3.8 ResetDriftCompensation()	33
7.5 GazeUtilityLibrary.Tracker.BaseTracker Class Reference	33
7.5.1 Detailed Description	36
7.5.2 Constructor & Destructor Documentation	36
7.5.2.1 BaseTracker()	37
7.5.3 Member Function Documentation	38
7.5.3.1 ApplyCalibration()	38
7.5.3.2 CollectCalibrationDataAsync()	38
7.5.3.3 CollectValidationDataAsync()	39
7.5.3.4 ComputeValidation()	40
7.5.3.5 Dispose() [1/2]	40
7.5.3.6 Dispose() [2/2]	40
7.5.3.7 DriftCompensationEventHandler()	41
7.5.3.8 FinishCalibration()	41
7.5.3.9 FinishCalibrationAsync()	41
7.5.3.10 FinishValidation()	41
7.5.3.11 GazeDataHandler()	41
7.5.3.12 GetFixationFrameCount()	42
7.5.3.13 GetUnitDirection()	42
7.5.3.14 InitCalibration()	42
7.5.3.15 InitCalibrationAsync()	43
7.5.3.16 InitDriftCompensation()	43
7.5.3.17 InitValidation()	43

7.5.3.18 IsInitialised()	43
7.5.3.19 IsReady()	44
7.5.3.20 OnGazeDataReceived()	44
7.5.3.21 OnPropertyChanged()	44
7.5.3.22 OnTrackerDisabled()	44
7.5.3.23 OnTrackerDisabledTimeout()	45
7.5.3.24 OnTrackerEnabled()	45
7.5.3.25 OnUserPositionDataReceived()	45
7.5.3.26 PatternReplace()	45
7.5.3.27 ResetDriftCompensation()	46
7.5.3.28 StartDriftCompensation()	46
7.5.3.29 UserPositionDataHandler()	46
7.5.4 Member Data Documentation	46
7.5.4.1 config	46
7.5.4.2 DeviceName	47
7.5.4.3 dialogBoxTimer	47
7.5.4.4 driftCompensation	47
7.5.4.5 logger	47
7.5.4.6 screenArea	47
7.5.4.7 trackerMessageBox	47
7.5.5 Property Documentation	48
7.5.5.1 State	48
7.5.6 Event Documentation	48
7.5.6.1 DriftCompensationComputed	48
7.5.6.2 GazeDataReceived	48
7.5.6.3 PropertyChanged	48
7.5.6.4 TrackerDisabled	48
7.5.6.5 TrackerEnabled	49
7.5.6.6 UserPositionDataReceived	49
7.6 CustomCalibrationLibrary.Views.Calibration Class Reference	49
7.6.1 Detailed Description	50
7.7 CustomCalibrationLibrary.Commands.CalibrationCommand Class Reference	50
7.7.1 Detailed Description	51
7.8 GazeUtilityLibrary.CalibrationDataError Class Reference	51
7.8.1 Member Function Documentation	52
7.8.1.1 GetCalibrationDataErrorString()	52
7.9 CustomCalibrationLibrary.Views.CalibrationFailed Class Reference	52
7.9.1 Detailed Description	53
7.9.2 Constructor & Destructor Documentation	53
7.9.2.1 CalibrationFailed()	53
7.9.3 Property Documentation	54
7.9.3.1 CalibrationAbortCommand	54

---

7.9.3.2 CalibrationRestartCommand . . . . .	54
7.9.3.3 Error . . . . .	54
7.9.4 Event Documentation . . . . .	54
7.9.4.1 PropertyChanged . . . . .	54
7.10 CustomCalibrationLibrary.Views.CalibrationFrame Class Reference . . . . .	55
7.10.1 Detailed Description . . . . .	55
7.11 CustomCalibrationLibrary.Models.CalibrationModel Class Reference . . . . .	56
7.11.1 Detailed Description . . . . .	57
7.11.2 Member Function Documentation . . . . .	57
7.11.2.1 GazeDataCollected() . . . . .	57
7.11.2.2 InitCalibration() . . . . .	58
7.11.2.3 NextCalibrationPoint() . . . . .	58
7.11.2.4 RedoCalibrationPoint() . . . . .	58
7.11.2.5 SetCalibrationResult() . . . . .	58
7.11.2.6 UpdateGazePoint() . . . . .	58
7.11.3 Property Documentation . . . . .	59
7.11.3.1 CalibrationPoints . . . . .	59
7.11.3.2 Error . . . . .	59
7.11.3.3 GazePoint . . . . .	59
7.11.3.4 Index . . . . .	59
7.11.3.5 LastStatus . . . . .	59
7.11.3.6 Points . . . . .	60
7.11.3.7 Status . . . . .	60
7.11.3.8 UserPositionGuide . . . . .	60
7.11.4 Event Documentation . . . . .	60
7.11.4.1 CalibrationEvent . . . . .	60
7.12 GazeUtilityLibrary.DataStructs.CalibrationPoint Class Reference . . . . .	61
7.12.1 Detailed Description . . . . .	62
7.12.2 Property Documentation . . . . .	62
7.12.2.1 GazePositionAverage . . . . .	62
7.12.2.2 GazePositionLeft . . . . .	62
7.12.2.3 GazePositionRight . . . . .	62
7.12.2.4 HasData . . . . .	63
7.12.2.5 Index . . . . .	63
7.12.2.6 Position . . . . .	63
7.13 CustomCalibrationLibrary.Views.CalibrationPoint Class Reference . . . . .	63
7.13.1 Detailed Description . . . . .	64
7.14 CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel Class Reference . . . . .	64
7.14.1 Detailed Description . . . . .	65
7.15 CustomCalibrationLibrary.Views.CalibrationResult Class Reference . . . . .	65
7.15.1 Detailed Description . . . . .	66
7.16 CustomCalibrationLibrary.Views.CalibrationResultPoint Class Reference . . . . .	67

---

7.16.1 Detailed Description	67
7.17 CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel Class Reference	68
7.17.1 Detailed Description	69
7.17.2 Constructor & Destructor Documentation	69
7.17.2.1 CalibrationResultViewModel()	69
7.17.3 Member Function Documentation	69
7.17.3.1 OnGazeToggle()	69
7.17.4 Property Documentation	70
7.17.4.1 CalibrationAcceptCommand	70
7.17.4.2 CalibrationRestartCommand	70
7.17.4.3 GazePoint	70
7.17.4.4 GazeVisibilityCommand	70
7.18 Tobii.Research.Addons.CalibrationValidationPoint Class Reference	70
7.18.1 Detailed Description	71
7.18.2 Property Documentation	71
7.18.2.1 AccuracyLeftEye	71
7.18.2.2 AccuracyRightEye	71
7.18.2.3 Coordinates	72
7.18.2.4 GazeData	72
7.18.2.5 PrecisionLeftEye	72
7.18.2.6 PrecisionRightEye	72
7.18.2.7 PrecisionRMSLeftEye	72
7.18.2.8 PrecisionRMSRightEye	72
7.18.2.9 TimedOut	73
7.19 Tobii.Research.Addons.CalibrationValidationResult Class Reference	73
7.19.1 Detailed Description	73
7.19.2 Property Documentation	73
7.19.2.1 AverageAccuracyLeftEye	74
7.19.2.2 AverageAccuracyRightEye	74
7.19.2.3 AveragePrecisionLeftEye	74
7.19.2.4 AveragePrecisionRightEye	74
7.19.2.5 AveragePrecisionRMSLeftEye	74
7.19.2.6 AveragePrecisionRMSRightEye	74
7.19.2.7 Points	75
7.20 CustomCalibrationLibrary.ViewModels.CalibrationViewModel Class Reference	75
7.20.1 Detailed Description	76
7.20.2 Constructor & Destructor Documentation	76
7.20.2.1 CalibrationViewModel()	76
7.20.3 Property Documentation	76
7.20.3.1 CalibrationPoints	76
7.21 CustomCalibrationLibrary.Views.CalibrationWindow Class Reference	77
7.21.1 Detailed Description	77

7.22 CustomCalibrationLibrary.Views.Computing Class Reference . . . . .	78
7.22.1 Detailed Description . . . . .	78
7.23 GazeUtilityLibrary.ConfigItem Class Reference . . . . .	78
7.23.1 Detailed Description . . . . .	79
7.24 GazeUtilityLibrary.ConfigScreenArea Class Reference . . . . .	79
7.25 CustomCalibrationLibrary.Views.Disconnect Class Reference . . . . .	80
7.25.1 Detailed Description . . . . .	81
7.25.2 Property Documentation . . . . .	81
7.25.2.1 CalibrationAbortCommand . . . . .	81
7.26 GazeUtilityLibrary.DriftCompensation Class Reference . . . . .	81
7.26.1 Detailed Description . . . . .	82
7.26.2 Member Function Documentation . . . . .	82
7.26.2.1 Reset() . . . . .	82
7.26.2.2 Start() . . . . .	82
7.26.2.3 Update() . . . . .	82
7.26.3 Property Documentation . . . . .	83
7.26.3.1 Q . . . . .	83
7.27 GazeUtilityLibrary.DataStructs.DriftCompensationData Class Reference . . . . .	83
7.27.1 Detailed Description . . . . .	83
7.27.2 Constructor & Destructor Documentation . . . . .	83
7.27.2.1 DriftCompensationData() . . . . .	83
7.27.3 Property Documentation . . . . .	84
7.27.3.1 Compensation . . . . .	84
7.27.3.2 GazePosition2d . . . . .	84
7.27.3.3 GazePosition3d . . . . .	84
7.28 CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel Class Reference . . . . .	84
7.28.1 Detailed Description . . . . .	85
7.28.2 Constructor & Destructor Documentation . . . . .	85
7.28.2.1 DriftCompensationViewModel() . . . . .	85
7.28.3 Property Documentation . . . . .	85
7.28.3.1 FixationPoint . . . . .	85
7.29 CustomCalibrationLibrary.Views.DriftCompensationWindow Class Reference . . . . .	86
7.29.1 Detailed Description . . . . .	86
7.30 GazeUtilityLibrary.DataStructs.EyeData Class Reference . . . . .	86
7.30.1 Detailed Description . . . . .	87
7.30.2 Constructor & Destructor Documentation . . . . .	87
7.30.2.1 EyeData() . . . . .	87
7.30.3 Property Documentation . . . . .	87
7.30.3.1 IsPupilDiameterValid . . . . .	87
7.30.3.2 PupilDiameter . . . . .	88
7.31 GazeUtilityLibrary.Tracker.EyeTrackerPro Class Reference . . . . .	88
7.31.1 Detailed Description . . . . .	89



7.31.2 Constructor & Destructor Documentation	90
7.31.2.1 EyeTrackerPro()	90
7.31.3 Member Function Documentation	90
7.31.3.1 ApplyCalibration()	90
7.31.3.2 CollectCalibrationDataAsync()	90
7.31.3.3 CollectValidationDataAsync()	91
7.31.3.4 ComputeValidation()	91
7.31.3.5 FinishCalibration()	92
7.31.3.6 FinishCalibrationAsync()	92
7.31.3.7 FinishValidation()	92
7.31.3.8 GetFixationFrameCount()	92
7.31.3.9 GetUnitDirection()	93
7.31.3.10 InitCalibration()	93
7.31.3.11 InitCalibrationAsync()	93
7.31.3.12 InitDriftCompensation()	93
7.31.3.13 InitValidation()	94
7.31.3.14 IsInitialised()	94
7.31.3.15 IsLicenseOk()	94
7.31.3.16 PatternReplace()	94
7.32 CustomCalibrationLibrary.Views.FixationPoint Class Reference	95
7.32.1 Detailed Description	95
7.33 GazeUtilityLibrary.DataStructs.GazeCalibrationData Class Reference	95
7.33.1 Detailed Description	96
7.33.2 Constructor & Destructor Documentation	96
7.33.2.1 GazeCalibrationData()	96
7.33.3 Member Function Documentation	97
7.33.3.1 Prepare()	97
7.34 GazeUtilityLibrary.GazeConfigError Class Reference	97
7.34.1 Member Function Documentation	98
7.34.1.1 GetGazeConfigErrorString()	98
7.35 GazeUtilityLibrary.GazeConfiguration Class Reference	99
7.35.1 Member Function Documentation	99
7.35.1.1 CleanupCalibrationOutputFile()	99
7.35.1.2 CleanupGazeOutputFile()	100
7.35.1.3 CleanupValidationOutputFile()	100
7.35.1.4 DumpCurrentConfigurationFile()	100
7.35.1.5 InitConfig()	101
7.35.1.6 PrepareCalibrationOutputFile()	101
7.35.1.7 PrepareGazeOutputFile()	101
7.35.1.8 PrepareValidationOutputFile()	102
7.35.1.9 WriteToCalibrationOutput()	102
7.35.1.10 WriteToGazeOutput()	102

7.35.1.11 WriteToValidationOutput()	102
7.36 GazeUtilityLibrary.DataStructs.GazeData Class Reference	103
7.36.1 Detailed Description	103
7.36.2 Constructor & Destructor Documentation	103
7.36.2.1 GazeData() [1/3]	103
7.36.2.2 GazeData() [2/3]	104
7.36.2.3 GazeData() [3/3]	104
7.36.3 Member Function Documentation	105
7.36.3.1 Prepare()	105
7.37 GazeUtilityLibrary.DataStructs.GazeData2d Class Reference	106
7.37.1 Detailed Description	106
7.37.2 Constructor & Destructor Documentation	106
7.37.2.1 GazeData2d()	106
7.38 GazeUtilityLibrary.DataStructs.GazeData3d Class Reference	106
7.38.1 Detailed Description	107
7.38.2 Constructor & Destructor Documentation	107
7.38.2.1 GazeData3d()	107
7.39 GazeUtilityLibrary.DataStructs.GazeDataCollection Class Reference	107
7.39.1 Detailed Description	108
7.39.2 Constructor & Destructor Documentation	108
7.39.2.1 GazeDataCollection() [1/2]	108
7.39.2.2 GazeDataCollection() [2/2]	108
7.40 GazeUtilityLibrary.GazeDataError Class Reference	109
7.40.1 Member Function Documentation	110
7.40.1.1 GetGazeDataErrorString()	110
7.41 GazeUtilityLibrary.GazeError Class Reference	111
7.41.1 Member Function Documentation	111
7.41.1.1 ConvertToBinString()	111
7.42 GazeUtilityLibrary.DataStructs.GazeValidationData Class Reference	112
7.42.1 Member Function Documentation	112
7.42.1.1 Prepare()	112
7.43 CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter Class Reference	112
7.43.1 Detailed Description	113
7.44 GazeUtilityLibrary.JsonConfigParser Class Reference	114
7.44.1 Detailed Description	114
7.44.2 Constructor & Destructor Documentation	114
7.44.2.1 JsonConfigParser()	114
7.44.3 Member Function Documentation	114
7.44.3.1 GetDefaultConfig()	114
7.44.3.2 ParseJsonConfig()	115
7.44.3.3 SerializeJsonConfig()	115
7.45 GazeUtilityLibrary.DataStructs.LiveGazePoint Class Reference	116

7.46 CustomCalibrationLibrary.ViewModels.Monitor Class Reference . . . . .	117
7.47 GazeUtilityLibrary.MouseHider Class Reference . . . . .	117
7.47.1 Detailed Description . . . . .	117
7.47.2 Constructor & Destructor Documentation . . . . .	117
7.47.2.1 MouseHider() . . . . .	117
7.47.3 Member Function Documentation . . . . .	118
7.47.3.1 HideCursor() . . . . .	118
7.47.3.2 ShowCursor() . . . . .	118
7.48 GazeUtilityLibrary.Tracker.MouseTracker Class Reference . . . . .	118
7.48.1 Detailed Description . . . . .	120
7.48.2 Constructor & Destructor Documentation . . . . .	120
7.48.2.1 MouseTracker() . . . . .	120
7.48.3 Member Function Documentation . . . . .	121
7.48.3.1 ApplyCalibration() . . . . .	121
7.48.3.2 CollectCalibrationDataAsync() . . . . .	121
7.48.3.3 CollectValidationDataAsync() . . . . .	121
7.48.3.4 ComputeValidation() . . . . .	122
7.48.3.5 Dispose() . . . . .	122
7.48.3.6 FinishCalibration() . . . . .	122
7.48.3.7 FinishCalibrationAsync() . . . . .	123
7.48.3.8 FinishValidation() . . . . .	123
7.48.3.9 GetFixationFrameCount() . . . . .	123
7.48.3.10 GetUnitDirection() . . . . .	123
7.48.3.11 InitCalibration() . . . . .	124
7.48.3.12 InitCalibrationAsync() . . . . .	124
7.48.3.13 InitDriftCompensation() . . . . .	124
7.48.3.14 InitValidation() . . . . .	124
7.48.3.15 Start() . . . . .	124
7.48.3.16 Stop() . . . . .	125
7.49 GazeUtilityLibrary.DataStructs.PipeCommand Class Reference . . . . .	125
7.50 CustomCalibrationLibrary.Converters.PositionConverter Class Reference . . . . .	125
7.50.1 Detailed Description . . . . .	126
7.50.2 Member Data Documentation . . . . .	126
7.50.2.1 OffsetProperty . . . . .	126
7.51 CustomCalibrationLibrary.Converters.ProximityColorConverter Class Reference . . . . .	127
7.51.1 Detailed Description . . . . .	127
7.52 GazeUtilityLibrary.ScreenArea Class Reference . . . . .	128
7.52.1 Detailed Description . . . . .	128
7.52.2 Constructor & Destructor Documentation . . . . .	128
7.52.2.1 ScreenArea() . . . . .	128
7.52.3 Member Function Documentation . . . . .	129
7.52.3.1 Dump() . . . . .	129

7.52.3.2 GetIntersectionPoint()	129
7.52.3.3 GetPoint2d()	130
7.52.3.4 GetPoint2dNormalized()	130
7.53 Tobii.Research.Addons.ScreenBasedCalibrationValidation Class Reference	130
7.53.1 Detailed Description	132
7.53.2 Member Enumeration Documentation	132
7.53.2.1 ValidationState	132
7.53.3 Constructor & Destructor Documentation	132
7.53.3.1 ScreenBasedCalibrationValidation()	132
7.53.4 Member Function Documentation	133
7.53.4.1 Compute()	133
7.53.4.2 DiscardData()	133
7.53.4.3 Dispose()	133
7.53.4.4 EnterValidationMode()	134
7.53.4.5 LeaveValidationMode()	134
7.53.4.6 StartCollectingData()	134
7.53.5 Property Documentation	134
7.53.5.1 Result	134
7.53.5.2 State	134
7.54 CustomCalibrationLibrary.Views.ScreenSelection Class Reference	135
7.54.1 Detailed Description	135
7.55 CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel Class Reference	136
7.55.1 Property Documentation	136
7.55.1.1 CalibrationAbortCommand	136
7.55.1.2 CalibrationStartCommand	136
7.56 GazeUtilityLibrary.ScreenTriangle Class Reference	136
7.56.1 Detailed Description	137
7.56.2 Member Function Documentation	137
7.56.2.1 GetIntersectionPoint()	137
7.57 GazeUtilityLibrary.TrackerLogger Class Reference	137
7.57.1 Detailed Description	138
7.57.2 Constructor & Destructor Documentation	138
7.57.2.1 TrackerLogger()	138
7.57.3 Member Function Documentation	138
7.57.3.1 Debug()	138
7.57.3.2 DumpFatal()	139
7.57.3.3 Error()	139
7.57.3.4 Info()	139
7.57.3.5 Warning()	139
7.58 GazeUtilityLibrary.TrackerMessageBox Class Reference	140
7.58.1 Detailed Description	140
7.59 GazeUtilityLibrary.DataStructs.UserPositionData Class Reference	141

7.60 CustomCalibrationLibrary.Views.UserPositionGuide Class Reference . . . . .	142
7.60.1 Detailed Description . . . . .	143
7.61 CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel Class Reference . . . . .	143
7.61.1 Detailed Description . . . . .	143
7.61.2 Constructor & Destructor Documentation . . . . .	143
7.61.2.1 UserPositionGuideViewModel() . . . . .	143
7.61.3 Property Documentation . . . . .	144
7.61.3.1 CalibrationAbortCommand . . . . .	144
7.61.3.2 CalibrationStartCommand . . . . .	144
7.61.3.3 UserPosition . . . . .	144
7.62 CustomCalibrationLibrary.Views.ValidationResult Class Reference . . . . .	145
7.62.1 Detailed Description . . . . .	145
7.63 CustomCalibrationLibrary.ViewModels.ValidationResultViewModel Class Reference . . . . .	146
7.63.1 Detailed Description . . . . .	146
7.63.2 Constructor & Destructor Documentation . . . . .	146
7.63.2.1 ValidationResultViewModel() . . . . .	146
7.63.3 Property Documentation . . . . .	146
7.63.3.1 ValidationCloseCommand . . . . .	147
7.63.3.2 ValidationData . . . . .	147
7.63.3.3 ValidationRestartCommand . . . . .	147
<b>Index</b>	<b>149</b>



# Chapter 1

## v3.2.1

### 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 output data.
- Add pipe command `SET_TAG` to allow annotate data samples.
- Add pipe command `RESET_START_TIMER` to reset 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 separate 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 field '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 startup
  - 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](#).

## Installation

The complete toolset package can be downloaded from the [release folder](#). 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 has 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 TERMINATE\*\*** should be used.
- **\*\*GazeControl.exe\*\*** This program allows to interact with **\*\*Gaze.exe\*\*** by passing the arguments `/command <COMMAND>`, `/value <VALUE>`, and `/reset` to the application. Passing an argument to an application can be done in command line or by creating a shortcut to the program. Corresponding shortcuts for all available `<COMMAND>`s are provided in the release package. The following `<COMMAND>`s are available (use argument `/value <VALUE>` whenever a command accepts a value):
  - `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](#).
  - `VALIDATE` uses the [Tobii Pro SDK Addon](#) and launches a validation process.
  - `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.
  - 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 `DRI↔FT_COMPENSATION`.
  - TERMINATE requests `**Gaze.exe**` to close gracefully and logs these events to the log file.
  - SET\_TAG <TAG> sets a custom tag <TAG> which will be added to each data sample in the output file (use argument `/value` to define the <TAG>).
  - SET\_TRIAL\_ID <ID> sets a trial ID integer number <ID> which will be added to each data sample in the output file (use argument `/value` to define the <ID>).
  - RESET\_START\_TIME allows to reset the relative timestamp. The relative timestamp can also be reset by passing the argument `/reset` to the application with any of the above commands.
- `**ShowMouse.exe**` This program allows to restore the standard mouse pointer. It might be useful if the program `\texttt{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 `\texttt{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`

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 processes:

- [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`

Note 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 information 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

To start a process from within opensesame use a python script. The following example starts the custom calibration program with the subject number passed as argument:

```
import subprocess
subprocess.run(["CustomCalibrate.exe", "/subject", var.get(u'subject_nr')])
```

## Release Notes

Information about the releases can be found in the [CHANGELOG](#)





## Chapter 3

# Namespace Index

### 3.1 Namespace List

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

<a href="#">CustomCalibrationLibrary</a> . . . . .	19
<a href="#">CustomCalibrationLibrary.Commands</a> . . . . .	19
<a href="#">CustomCalibrationLibrary.Converters</a> . . . . .	19
<a href="#">CustomCalibrationLibrary.Models</a> . . . . .	19
<a href="#">CustomCalibrationLibrary.ViewModels</a> . . . . .	20
<a href="#">CustomCalibrationLibrary.Views</a> . . . . .	21
<a href="#">GazeControl</a> . . . . .	21
<a href="#">GazeToMouse</a> . . . . .	21
<a href="#">GazeUtilityLibrary</a> helper class to show and hide the system cursor . . . . .	22
<a href="#">GazeUtilityLibrary.DataStructs</a> . . . . .	23
<a href="#">GazeUtilityLibrary.Tracker</a> . . . . .	25
<a href="#">ShowMouse</a> . . . . .	25
<a href="#">Tobii</a> . . . . .	25
<a href="#">Tobii.Research</a> . . . . .	25
<a href="#">Tobii.Research.Addons</a> . . . . .	25
<a href="#">Tobii.Research.Addons.Utility</a> . . . . .	26
<a href="#">TobiiCalibrate</a> . . . . .	26



## Chapter 4

# Hierarchical Index

### 4.1 Class Hierarchy

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

Application	
GazeControl.App . . . . .	27
GazeToMouse.App . . . . .	30
ShowMouse.App . . . . .	28
TobiiCalibrate.App . . . . .	29
Tobii.Research.Addons.CalibrationValidationPoint . . . . .	70
Tobii.Research.Addons.CalibrationValidationResult . . . . .	73
CustomCalibrationLibrary.ViewModels.CalibrationViewModel . . . . .	75
CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel . . . . .	68
GazeUtilityLibrary.ConfigItem . . . . .	78
GazeUtilityLibrary.ConfigScreenArea . . . . .	79
DependencyObject	
CustomCalibrationLibrary.Converters.PositionConverter . . . . .	125
GazeUtilityLibrary.DriftCompensation . . . . .	81
GazeUtilityLibrary.DataStructs.DriftCompensationData . . . . .	83
CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel . . . . .	84
GazeUtilityLibrary.DataStructs.EyeData . . . . .	86
Frame	
CustomCalibrationLibrary.Views.CalibrationFrame . . . . .	55
GazeUtilityLibrary.DataStructs.GazeCalibrationData . . . . .	95
GazeUtilityLibrary.GazeConfiguration . . . . .	99
GazeUtilityLibrary.DataStructs.GazeData . . . . .	103
GazeUtilityLibrary.DataStructs.GazeData2d . . . . .	106
GazeUtilityLibrary.DataStructs.GazeData3d . . . . .	106
GazeUtilityLibrary.DataStructs.GazeDataCollection . . . . .	107
GazeUtilityLibrary.GazeError . . . . .	111
GazeUtilityLibrary.CalibrationDataError . . . . .	51
GazeUtilityLibrary.GazeConfigError . . . . .	97
GazeUtilityLibrary.GazeDataError . . . . .	109
GazeUtilityLibrary.DataStructs.GazeValidationData . . . . .	112
ICommand	
CustomCalibrationLibrary.Commands.CalibrationCommand . . . . .	50
IDisposable	
GazeUtilityLibrary.Tracker.BaseTracker . . . . .	33
GazeUtilityLibrary.Tracker.EyeTrackerPro . . . . .	88

GazeUtilityLibrary.Tracker.MouseTracker . . . . .	118
Tobii.Research.Addons.ScreenBasedCalibrationValidation . . . . .	130
INotifyPropertyChanged	
CustomCalibrationLibrary.Models.CalibrationModel . . . . .	56
CustomCalibrationLibrary.Views.CalibrationFailed . . . . .	52
GazeUtilityLibrary.DataStructs.CalibrationPoint . . . . .	61
CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel . . . . .	64
GazeUtilityLibrary.DataStructs.LiveGazePoint . . . . .	116
GazeUtilityLibrary.DataStructs.UserPositionData . . . . .	141
GazeUtilityLibrary.Tracker.BaseTracker . . . . .	33
IValueConverter	
CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter . . . . .	112
CustomCalibrationLibrary.Converters.PositionConverter . . . . .	125
CustomCalibrationLibrary.Converters.ProximityColorConverter . . . . .	127
GazeUtilityLibrary.JsonConfigParser . . . . .	114
CustomCalibrationLibrary.ViewModels.Monitor . . . . .	117
GazeUtilityLibrary.MouseHider . . . . .	117
Page	
CustomCalibrationLibrary.Views.Calibration . . . . .	49
CustomCalibrationLibrary.Views.CalibrationFailed . . . . .	52
CustomCalibrationLibrary.Views.Computing . . . . .	78
CustomCalibrationLibrary.Views.Disconnect . . . . .	80
Page	
CustomCalibrationLibrary.Views.CalibrationResult . . . . .	65
CustomCalibrationLibrary.Views.ScreenSelection . . . . .	135
CustomCalibrationLibrary.Views.UserPositionGuide . . . . .	142
CustomCalibrationLibrary.Views.ValidationResult . . . . .	145
GazeUtilityLibrary.DataStructs.PipeCommand . . . . .	125
GazeUtilityLibrary.ScreenArea . . . . .	128
CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel . . . . .	136
GazeUtilityLibrary.ScreenTriangle . . . . .	136
GazeUtilityLibrary.TrackerLogger . . . . .	137
UserControl	
CustomCalibrationLibrary.Views.CalibrationPoint . . . . .	63
CustomCalibrationLibrary.Views.CalibrationResultPoint . . . . .	67
CustomCalibrationLibrary.Views.FixationPoint . . . . .	95
CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel . . . . .	143
CustomCalibrationLibrary.ViewModels.ValidationResultViewModel . . . . .	146
Window	
CustomCalibrationLibrary.Views.CalibrationWindow . . . . .	77
CustomCalibrationLibrary.Views.DriftCompensationWindow . . . . .	86
GazeUtilityLibrary.TrackerMessageBox . . . . .	140

## Chapter 5

# Class Index

### 5.1 Class List

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

<a href="#">GazeControl.App</a>	27
Interaction logic for App.xaml . . . . .	
<a href="#">ShowMouse.App</a>	28
Interaction logic for App.xaml . . . . .	
<a href="#">TobiiCalibrate.App</a>	29
Interaction logic for App.xaml . . . . .	
<a href="#">GazeToMouse.App</a>	30
Interaction logic for App.xaml . . . . .	
<a href="#">GazeUtilityLibrary.Tracker.BaseTracker</a>	33
The common interface for the <a href="#">Tobii</a> eyetracker Engines Core and Pro . . . . .	
<a href="#">CustomCalibrationLibrary.Views.Calibration</a>	49
Interaction logic for Calibration.xaml . . . . .	
<a href="#">CustomCalibrationLibrary.Commands.CalibrationCommand</a>	50
Comand class to trigger calibration events . . . . .	
<a href="#">GazeUtilityLibrary.CalibrationDataError</a>	51
<a href="#">CustomCalibrationLibrary.Views.CalibrationFailed</a>	52
Interaction logic for CalibrationFailed.xaml . . . . .	
<a href="#">CustomCalibrationLibrary.Views.CalibrationFrame</a>	55
Interaction logic for CalibrationCollection.xaml . . . . .	
<a href="#">CustomCalibrationLibrary.Models.CalibrationModel</a>	56
The model for the calibration process . . . . .	
<a href="#">GazeUtilityLibrary.DataStructs.CalibrationPoint</a>	61
A calibration point class holding several metrics connected to a calibration point . . . . .	
<a href="#">CustomCalibrationLibrary.Views.CalibrationPoint</a>	63
Interaction logic for CalibrationPoint.xaml . . . . .	
<a href="#">CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel</a>	64
The view model for a calibration point . . . . .	
<a href="#">CustomCalibrationLibrary.Views.CalibrationResult</a>	65
Interaction logic for CalibrationResult.xaml . . . . .	
<a href="#">CustomCalibrationLibrary.Views.CalibrationResultPoint</a>	67
Interaction logic for CalibrationResultPoint.xaml . . . . .	
<a href="#">CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel</a>	68
View model class of the gaze calibration result . . . . .	
<a href="#">Tobii.Research.Addons.CalibrationValidationPoint</a>	70
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 . . . . .	

<a href="#">Tobii.Research.Addons.CalibrationValidationResult</a>	
Contains the result of the calibration validation . . . . .	73
<a href="#">CustomCalibrationLibrary.ViewModels.CalibrationViewModel</a>	
The view model class of the calibration view . . . . .	75
<a href="#">CustomCalibrationLibrary.Views.CalibrationWindow</a>	
Interaction logic for MainWindow.xaml . . . . .	77
<a href="#">CustomCalibrationLibrary.Views.Computing</a>	
Interaction logic for Computing.xaml . . . . .	78
<a href="#">GazeUtilityLibrary.ConfigItem</a>	
configuration file class . . . . .	78
<a href="#">GazeUtilityLibrary.ConfigScreenArea</a>	
. . . . .	79
<a href="#">CustomCalibrationLibrary.Views.Disconnect</a>	
Interaction logic for Disconnect.xaml . . . . .	80
<a href="#">GazeUtilityLibrary.DriftCompensation</a>	
The class to handle drift compensation . . . . .	81
<a href="#">GazeUtilityLibrary.DataStructs.DriftCompensationData</a>	
The drift compensation data structure . . . . .	83
<a href="#">CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel</a>	
The view model class of the drift compensation view . . . . .	84
<a href="#">CustomCalibrationLibrary.Views.DriftCompensationWindow</a>	
Interaction logic for DriftCompensation.xaml . . . . .	86
<a href="#">GazeUtilityLibrary.DataStructs.EyeData</a>	
The eye data set, including pupil information . . . . .	86
<a href="#">GazeUtilityLibrary.Tracker.EyeTrackerPro</a>	
Interface to the <a href="#">Tobii</a> SDK Pro engine . . . . .	88
<a href="#">CustomCalibrationLibrary.Views.FixationPoint</a>	
Interaction logic for FixationPoint.xaml . . . . .	95
<a href="#">GazeUtilityLibrary.DataStructs.GazeCalibrationData</a>	
The event argument class for <a href="#">Tobii</a> eyetracker data . . . . .	95
<a href="#">GazeUtilityLibrary.GazeConfigError</a>	
. . . . .	97
<a href="#">GazeUtilityLibrary.GazeConfiguration</a>	
. . . . .	99
<a href="#">GazeUtilityLibrary.DataStructs.GazeData</a>	
The class definition of a gaze data set . . . . .	103
<a href="#">GazeUtilityLibrary.DataStructs.GazeData2d</a>	
The 2d gaze data set . . . . .	106
<a href="#">GazeUtilityLibrary.DataStructs.GazeData3d</a>	
The 3d gaze data set . . . . .	106
<a href="#">GazeUtilityLibrary.DataStructs.GazeDataCollection</a>	
The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data . . .	107
<a href="#">GazeUtilityLibrary.GazeDataError</a>	
. . . . .	109
<a href="#">GazeUtilityLibrary.GazeError</a>	
. . . . .	111
<a href="#">GazeUtilityLibrary.DataStructs.GazeValidationData</a>	
. . . . .	112
<a href="#">CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter</a>	
Converts True to Hidden and False to Visible . . . . .	112
<a href="#">GazeUtilityLibrary.JsonConfigParser</a>	
The config file "config.json" is parsed and its values are attributed to the <a href="#">ConfigItem</a> class . . .	114
<a href="#">GazeUtilityLibrary.DataStructs.LiveGazePoint</a>	
. . . . .	116
<a href="#">CustomCalibrationLibrary.ViewModels.Monitor</a>	
. . . . .	117
<a href="#">GazeUtilityLibrary.MouseHider</a>	
hide standard mouse pointer and restore it . . . . .	117
<a href="#">GazeUtilityLibrary.Tracker.MouseTracker</a>	
This class is used to hook into the system mouse events and track the position . . . . .	118
<a href="#">GazeUtilityLibrary.DataStructs.PipeCommand</a>	
. . . . .	125
<a href="#">CustomCalibrationLibrary.Converters.PositionConverter</a>	
Converter class to convert a normalized coordinate to a pixel coordinate . . . . .	125
<a href="#">CustomCalibrationLibrary.Converters.ProximityColorConverter</a>	
Converter class to convert the proximo of a normalized coordinate to the center point (0.5) into colors . . . . .	127

<a href="#">GazeUtilityLibrary.ScreenArea</a>	
The class describing the Screen area in 3d and 2d space . . . . .	128
<a href="#">Tobii.Research.Addons.ScreenBasedCalibrationValidation</a>	
Provides methods and properties for managing calibration validation for screen based eye track- ers . . . . .	130
<a href="#">CustomCalibrationLibrary.Views.ScreenSelection</a>	
Interaction logic for ScreenSelection.xaml . . . . .	135
<a href="#">CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel</a>	
. . . . .	136
<a href="#">GazeUtilityLibrary.ScreenTriangle</a>	
A class to describe a triangle. This was supposed to be used to construct the <a href="#">ScreenArea</a> 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 . . . . .	136
<a href="#">GazeUtilityLibrary.TrackerLogger</a>	
Simple logger class . . . . .	137
<a href="#">GazeUtilityLibrary.TrackerMessageBox</a>	
Interaction logic for TrackerMessageBox.xaml . . . . .	140
<a href="#">GazeUtilityLibrary.DataStructs.UserPositionData</a>	
. . . . .	141
<a href="#">CustomCalibrationLibrary.Views.UserPositionGuide</a>	
Interaction logic for UserPositionGuide.xaml . . . . .	142
<a href="#">CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel</a>	
The view model class for the user position guide view . . . . .	143
<a href="#">CustomCalibrationLibrary.Views.ValidationResult</a>	
Interaction logic for ValidationResult.xaml . . . . .	145
<a href="#">CustomCalibrationLibrary.ViewModels.ValidationResultViewModel</a>	
View model class of the gaze validation result . . . . .	146





## Chapter 6

# Namespace Documentation

### 6.1 CustomCalibrationLibrary Namespace Reference

### 6.2 CustomCalibrationLibrary.Commands Namespace Reference

#### Classes

- class [CalibrationCommand](#)  
*Comand class to trigger calibration events.*
- class **GazeVisibilityCommand**  
*Command class to change the gaze visibility*

### 6.3 CustomCalibrationLibrary.Converters Namespace Reference

#### Classes

- class [HasDataToVisibilityConverter](#)  
*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.*

### 6.4 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.*

### 6.4.1 Enumeration Type Documentation

#### 6.4.1.1 CalibrationEventType

enum [CustomCalibrationLibrary.Models.CalibrationEventType](#) [strong]

Events to trigger changes in the calibration process.

#### 6.4.1.2 CalibrationStatus

enum [CustomCalibrationLibrary.Models.CalibrationStatus](#) [strong]

The status of the calibarion process.

## 6.5 CustomCalibrationLibrary.ViewModels Namespace Reference

### Classes

- 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 [DriftCompensationViewModel](#)  
    *The view model class of the drift compensation view.*
- class [Monitor](#)
- class [ScreenSelectionViewModel](#)
- class [UserPositionGuideViewModel](#)  
    *The view model class for the user position guide view.*
- class [ValidationResultViewModel](#)  
    *View model class of the gaze validation result.*

## 6.6 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 [CalibrationResultPoint](#)  
*Interaction logic for CalibrationResultPoint.xaml*
- class [CalibrationWindow](#)  
*Interaction logic for MainWindow.xaml*
- class [Computing](#)  
*Interaction logic for Computing.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 [UserPositionGuide](#)  
*Interaction logic for UserPositionGuide.xaml*
- class [ValidationResult](#)  
*Interaction logic for ValidationResult.xaml*

## 6.7 GazeControl Namespace Reference

### Classes

- class [App](#)  
*Interaction logic for App.xaml*
- class **NamedPipeClient**

## 6.8 GazeToMouse Namespace Reference

### Classes

- class [App](#)  
*Interaction logic for App.xaml*

## 6.9 GazeUtilityLibrary Namespace Reference

helper class to show and hide the system cursor

### Classes

- class [CalibrationDataError](#)
- class **ConfigChecker**
- class [ConfigItem](#)  
*configuration file class*
- class [ConfigScreenArea](#)
- class [DriftCompensation](#)  
*The class to handle drift compensation.*
- class [GazeConfigError](#)
- class [GazeConfiguration](#)
- class [GazeDataError](#)
- class [GazeError](#)
- 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 restore 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 }
- enum [EGazeConfigError](#) {  
**FallbackToDefaultConfigName** = 0x001, **FallbackToCurrentOutputDir** = 0x002, **FallbackToDefault↵**  
**Config** = 0x004, **FallbackToDefaultDiameterFormat** = 0x008,  
**FallbackToDefaultOriginFormat** = 0x010, **FallbackToDefaultTimestampFormat** = 0x020, **OmitColumn↵**  
**Titles** = 0x040, **FallbackToDefaultColumnOrder** = 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*

#### 6.9.1 Detailed Description

helper class to show and hide the system cursor

## 6.9.2 Enumeration Type Documentation

### 6.9.2.1 ECalibrationDataError

enum [GazeUtilityLibrary.ECalibrationDataError](#) [strong]

Error values of the gaze output data

### 6.9.2.2 EGazeConfigError

enum [GazeUtilityLibrary.EGazeConfigError](#) [strong]

Error values of the configuration

### 6.9.2.3 EGazeDataError

enum [GazeUtilityLibrary.EGazeDataError](#) [strong]

Error values of the gaze output data

## 6.10 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 event argument class for [Tobii](#) eyetracker data*
- 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**
- class [GazeValidationData](#)
- class [LiveGazePoint](#)
- class [PipeCommand](#)
- class [UserPositionData](#)

## Enumerations

- enum [GazeOutputValue](#) {  
**DateTimeStamp = 0, DateTimeStampRelative, TrialId, Tag,**  
**CombinedGazePoint2dCompensatedX, CombinedGazePoint2dCompensatedY, CombinedGazePoint2dX, CombinedGazePoint2dY,**  
**CombinedGazePoint2dIsValid, CombinedGazePoint3dCompensatedX, CombinedGazePoint3dCompensatedY, CombinedGazePoint3dCompensatedZ,**  
**CombinedGazePoint3dX, CombinedGazePoint3dY, CombinedGazePoint3dZ, CombinedGazePoint3dIsValid,**  
**CombinedGazeOrigin3dX, CombinedGazeOrigin3dY, CombinedGazeOrigin3dZ, CombinedGazeOrigin3dIsValid,**  
**CombinedGazeDistance, CombinedPupilDiameter, CombinedPupilDiameterIsValid, 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 }**  
*enumerates output values produced by the eyetracker*
- enum [CalibrationOutputValue](#) {  
**Point2dX, Point2dY, LeftGazePoint2dX, LeftGazePoint2dY,**  
**LeftGazePoint2dIsValid, RightGazePoint2dX, RightGazePoint2dY, RightGazePoint2dIsValid }**  
*enumerates output values produced by the eyetracker*
- enum [ValidationOutputValue](#) {  
**LeftAccuracy, LeftPrecision, LeftPrecisionRMS, RightAccuracy,**  
**RightPrecision, RightPrecisionRMS }**  
*enumerates output values produced by the eyetracker*

### 6.10.1 Enumeration Type Documentation

#### 6.10.1.1 CalibrationOutputValue

enum [GazeUtilityLibrary.DataStructs.CalibrationOutputValue](#) [strong]

enumerates output values produced by the eyetracker

#### 6.10.1.2 GazeOutputValue

enum [GazeUtilityLibrary.DataStructs.GazeOutputValue](#) [strong]

enumerates output values produced by the eyetracker

### 6.10.1.3 ValidationOutputValue

enum [GazeUtilityLibrary.DataStructs.ValidationOutputValue](#) [strong]

enumerates output values produced by the eyetracker

## 6.11 GazeUtilityLibrary.Tracker Namespace Reference

### Classes

- class [BaseTracker](#)  
*The common interface for the [Tobii](#) eyetracker Engines Core and Pro*
- 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*

## 6.12 ShowMouse Namespace Reference

### Classes

- class [App](#)  
*Interaction logic for App.xaml*

## 6.13 Tobii Namespace Reference

## 6.14 Tobii.Research Namespace Reference

## 6.15 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.*

## 6.16 Tobii.Research.Addons.Utility Namespace Reference

### Classes

- class **Extensions**

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

- class **TimeKeeper**

## 6.17 TobiiCalibrate Namespace Reference

### Classes

- class [App](#)

*Interaction logic for App.xaml*



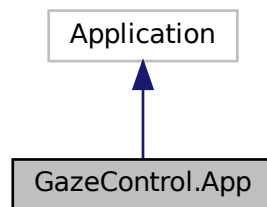
## Chapter 7

# Class Documentation

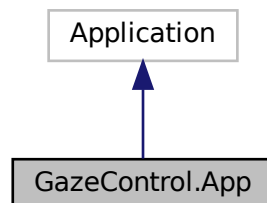
### 7.1 GazeControl.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for GazeControl.App:



Collaboration diagram for GazeControl.App:



### 7.1.1 Detailed Description

Interaction logic for App.xaml

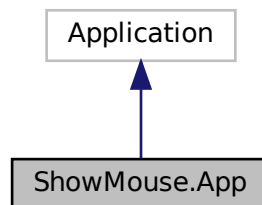
The documentation for this class was generated from the following file:

- `source/GazeControl/App.xaml.cs`

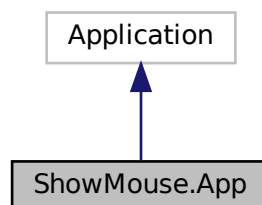
## 7.2 ShowMouse.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for ShowMouse.App:



Collaboration diagram for ShowMouse.App:



### 7.2.1 Detailed Description

Interaction logic for App.xaml

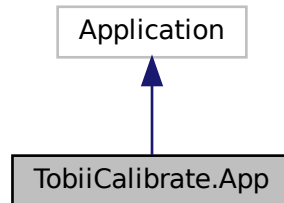
The documentation for this class was generated from the following file:

- `source/ShowMouse/App.xaml.cs`

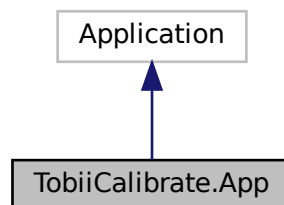
## 7.3 TobiiCalibrate.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for TobiiCalibrate.App:



Collaboration diagram for TobiiCalibrate.App:



### 7.3.1 Detailed Description

Interaction logic for App.xaml

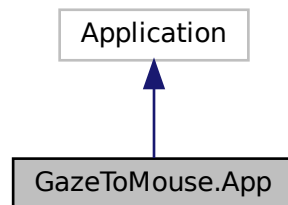
The documentation for this class was generated from the following file:

- `source/TobiiCalibrate/App.xaml.cs`

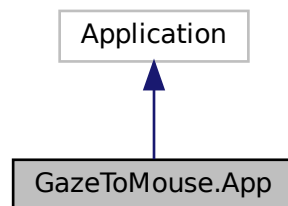
## 7.4 GazeToMouse.App Class Reference

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.*
- async Task< bool > [CompensateDrift](#) ()  
*Start the drift compensation process*
- 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]
- string **Tag** [get, set]
- int **TrialId** [get, set]

### 7.4.1 Detailed Description

Interaction logic for App.xaml

### 7.4.2 Constructor & Destructor Documentation

#### 7.4.2.1 App()

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

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

### 7.4.3 Member Function Documentation

#### 7.4.3.1 CalibrationValidate()

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

Start the gaze calibration process

#### Returns

True on success, false on failure

#### 7.4.3.2 CompensateDrift()

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

Start the drift compensation process

##### Returns

True on success, false on failure

#### 7.4.3.3 CustomCalibrate()

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

Start the gaze calibration process

##### Returns

True on success, false on failure

#### 7.4.3.4 GazeRecordingDisable()

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

Disable gaze recordings.

#### 7.4.3.5 GazeRecordingEnable()

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

Enable gaze recordings to disk.

#### 7.4.3.6 MouseTrackingDisable()

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

Disable mouse tracking.

#### 7.4.3.7 MouseTrackingEnable()

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

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

#### 7.4.3.8 ResetDriftCompensation()

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

Reset the current drift compensation offset to zero.

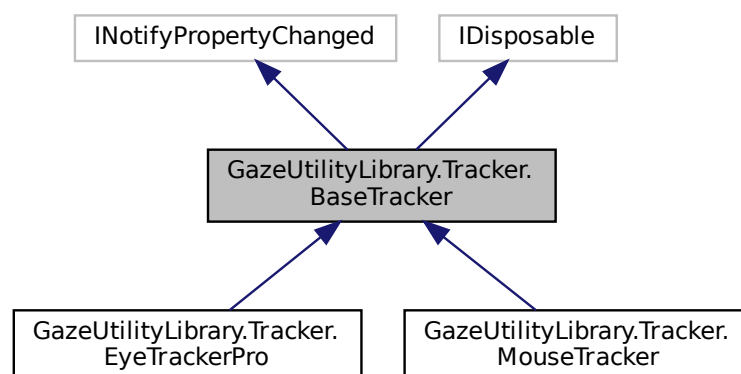
The documentation for this class was generated from the following file:

- source/GazeToMouse/App.xaml.cs

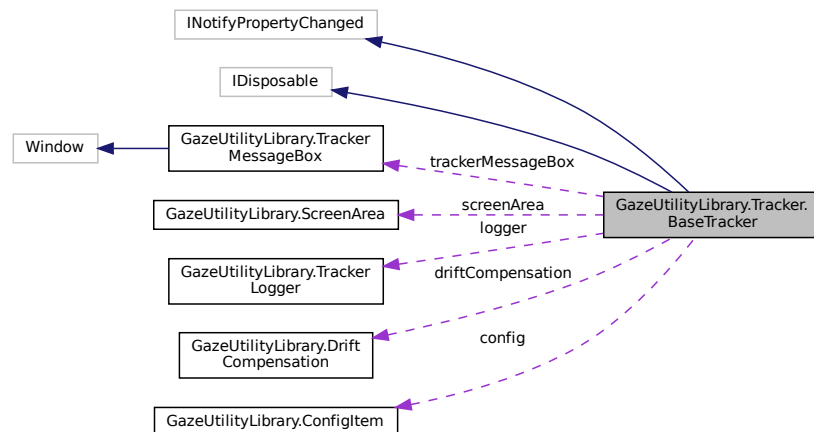
## 7.5 GazeUtilityLibrary.Tracker.BaseTracker Class Reference

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** }

## 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](#) ()  
*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 screen area structure holding the metrics of the screen in 3d space.*
- [ConfigItem](#) [config](#)  
*The gaze configuration item*

## Properties

- [ScreenArea?](#) **ScreenArea** [get]
- 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].*

### 7.5.1 Detailed Description

The common interface for the [Tobii](#) eyetracker Engines Core and Pro

See also

[INotifyPropertyChanged](#), [IDisposable](#)

### 7.5.2 Constructor & Destructor Documentation

### 7.5.2.1 BaseTracker()

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

Initializes a new instance of the EyeTrackerHandler class.

## Parameters

<i>logger</i>	The logger.
<i>ready_timer</i>	The ready timer.
<i>device_name</i>	Name of the device.

### 7.5.3 Member Function Documentation

#### 7.5.3.1 ApplyCalibration()

```
abstract Task<List<GazeCalibrationData> > GazeUtilityLibrary.Tracker.BaseTracker.ApplyCalibration ( ) [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](#).

#### 7.5.3.2 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

<i>point</i>	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](#).

### 7.5.3.3 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**

<i>point</i>	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](#).

**7.5.3.4 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](#).

**7.5.3.5 Dispose() [1/2]**

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

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

**7.5.3.6 Dispose() [2/2]**

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

Releases unmanaged and - optionally - managed resources.

**Parameters**

<i>disposing</i>	true to release both managed and unmanaged resources; false to release only unmanaged resources.
------------------	--

Reimplemented in [GazeUtilityLibrary.Tracker.MouseTracker](#).

### 7.5.3.7 DriftCompensationEventHandler()

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

Event handler for drift compensation events

#### Parameters

<i>sender</i>	The sender.
<i>driftCompensation</i>	The drift compensation quaternion

### 7.5.3.8 FinishCalibration()

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

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

Implemented in [GazeUtilityLibrary.Tracker.MouseTracker](#), and [GazeUtilityLibrary.Tracker.EyeTrackerPro](#).

### 7.5.3.9 FinishCalibrationAsync()

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

Finish the async calibration 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](#).

### 7.5.3.10 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](#).

### 7.5.3.11 GazeDataHandler()

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

Event handler for gaze data events of the eyetracker

## Parameters

<i>sender</i>	The sender.
<i>gazeData</i>	The e.

**7.5.3.12 GetFixationFrameCount()**

```
abstract int GazeUtilityLibrary.Tracker.BaseTracker.GetFixationFrameCount ( ) [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.

## Returns

The number of gaze samples to require for fixation detection.

Implemented in [GazeUtilityLibrary.Tracker.EyeTrackerPro](#), and [GazeUtilityLibrary.Tracker.MouseTracker](#).

**7.5.3.13 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](#).

**7.5.3.14 InitCalibration()**

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

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

Implemented in [GazeUtilityLibrary.Tracker.MouseTracker](#), and [GazeUtilityLibrary.Tracker.EyeTrackerPro](#).



#### 7.5.3.15 InitCalibrationAsync()

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

Initialise the async calibration 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](#).

#### 7.5.3.16 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](#).

#### 7.5.3.17 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](#).

#### 7.5.3.18 IsInitialised()

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

Checks whether 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](#).

#### 7.5.3.19 IsReady()

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

Determines whether this eye tracker is ready.

##### Returns

true if this instance is ready; otherwise, false.

#### 7.5.3.20 OnGazeDataReceived()

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

Called when [gaze data received].

##### Parameters

<i>data</i>	The gaze data event data.
-------------	---------------------------

#### 7.5.3.21 OnPropertyChanged()

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

Called when when the state property of EyeTracker is changing.

##### Parameters

<i>property_name</i>	Name of the property in WPF.
----------------------	------------------------------

#### 7.5.3.22 OnTrackerDisabled()

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

Raises the E:TrackerDisabled event.

##### Parameters

<i>e</i>	The EventArgs instance containing the event data.
----------	---

### 7.5.3.23 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

<i>source</i>	The source.
<i>e</i>	The ElapsedEventArgs instance containing the event data.

### 7.5.3.24 OnTrackerEnabled()

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

Raises the E:TrackerEnabled event.

#### Parameters

<i>e</i>	The EventArgs instance containing the event data.
----------	---

### 7.5.3.25 OnUserPositionDataReceived()

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

Called when [user position data received].

#### Parameters

<i>e</i>	The gaze data event data.
----------	---------------------------

### 7.5.3.26 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](#).

#### 7.5.3.27 ResetDriftCompensation()

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

Reset the drift compensation value

#### 7.5.3.28 StartDriftCompensation()

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

Start the drift compensation process.

#### 7.5.3.29 UserPositionDataHandler()

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

Event handler for user position data events of the eyetracker

#### Parameters

<i>sender</i>	The sender.
<i>e</i>	The e.

### 7.5.4 Member Data Documentation

#### 7.5.4.1 config

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

The gaze configuration item

#### 7.5.4.2 DeviceName

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

The name of the tracker device

#### 7.5.4.3 dialogBoxTimer

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

Timer to control the apperance of the dialog box

#### 7.5.4.4 driftCompensation

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

drift compensation handler

#### 7.5.4.5 logger

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

The logger

#### 7.5.4.6 screenArea

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

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

#### 7.5.4.7 trackerMessageBox

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

The dialog box that is controlled by the dialogBoxTimer

## 7.5.5 Property Documentation

### 7.5.5.1 State

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

Gets or sets the state of the eyetracker device.

The state.

## 7.5.6 Event Documentation

### 7.5.6.1 DriftCompensationComputed

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

Occurs when drift compensation was computed.

### 7.5.6.2 GazeDataReceived

`GazeDataHandler? GazeUtilityLibrary.Tracker.BaseTracker.GazeDataReceived`

Occurs when [gaze data received].

### 7.5.6.3 PropertyChanged

`PropertyChangedEventHandler? GazeUtilityLibrary.Tracker.BaseTracker.PropertyChanged`

Occurs when a property value changes.

### 7.5.6.4 TrackerDisabled

`EventHandler? GazeUtilityLibrary.Tracker.BaseTracker.TrackerDisabled`

Occurs when [tracker disabled].

### 7.5.6.5 TrackerEnabled

EventHandler? GazeUtilityLibrary.Tracker.BaseTracker.TrackerEnabled

Occurs when [tracker enabled].

### 7.5.6.6 UserPositionDataReceived

UserPositionDataHandler? GazeUtilityLibrary.Tracker.BaseTracker.UserPositionDataReceived

Occurs when [user position data received].

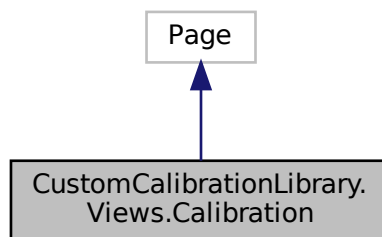
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Tracker/BaseTracker.cs

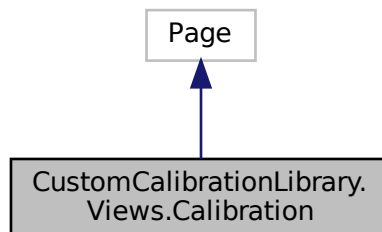
## 7.6 CustomCalibrationLibrary.Views.Calibration Class Reference

Interaction logic for Calibration.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.Calibration:



Collaboration diagram for CustomCalibrationLibrary.Views.Calibration:



## Public Member Functions

- **Calibration** ([CalibrationModel](#) model)

### 7.6.1 Detailed Description

Interaction logic for Calibration.xaml

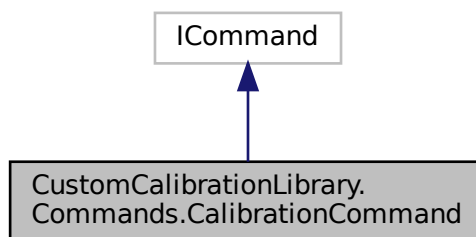
The documentation for this class was generated from the following file:

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

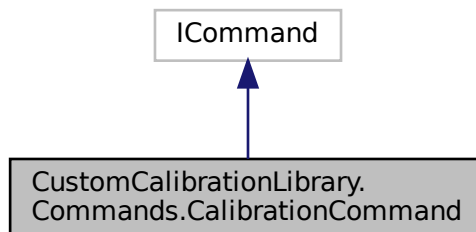
## 7.7 CustomCalibrationLibrary.Commands.CalibrationCommand Class Reference

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)
- bool **CanExecute** (object? parameter)
- void **Execute** (object? parameter)

## Properties

- EventHandler? **CanExecuteChanged**

### 7.7.1 Detailed Description

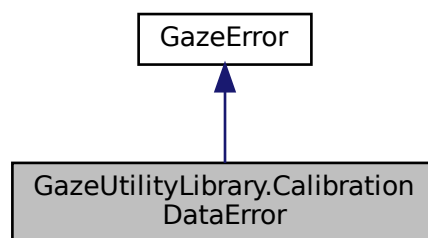
Comand class to trigger calibration events.

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

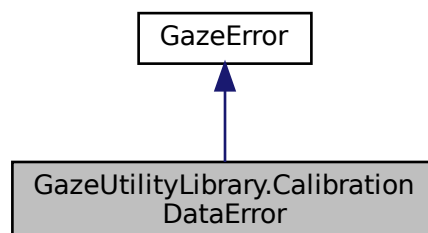
- `source/CustomCalibrationLibrary/Commands/CalibrationCommand.cs`

## 7.8 GazeUtilityLibrary.CalibrationDataError Class Reference

Inheritance diagram for GazeUtilityLibrary.CalibrationDataError:



Collaboration diagram for GazeUtilityLibrary.CalibrationDataError:



## Public Member Functions

- string [GetCalibrationDataErrorString](#) ()  
*Gets the gaze error string.*

## Properties

- [ECalibrationDataError](#) **Error** [set]

## Additional Inherited Members

### 7.8.1 Member Function Documentation

#### 7.8.1.1 GetCalibrationDataErrorString()

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

Gets the gaze error string.

#### Returns

the error string with binary error values if errors occurred, the empty string otherwise

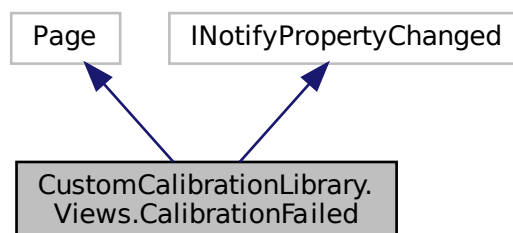
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeError.cs

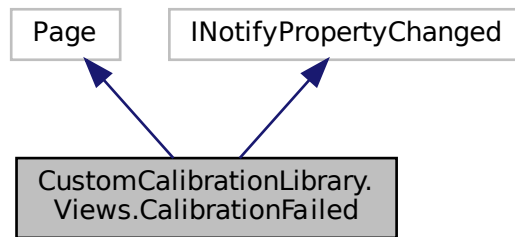
## 7.9 CustomCalibrationLibrary.Views.CalibrationFailed Class Reference

Interaction logic for CalibrationFailed.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationFailed:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationFailed:



## Public Member Functions

- [CalibrationFailed](#) ([CalibrationModel](#) model)  
*Constructor*

## 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.*

### 7.9.1 Detailed Description

Interaction logic for CalibrationFailed.xaml

### 7.9.2 Constructor & Destructor Documentation

#### 7.9.2.1 CalibrationFailed()

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

Constructor

**Parameters**

<i>model</i>	The claibration model
--------------	-----------------------

### 7.9.3 Property Documentation

#### 7.9.3.1 CalibrationAbortCommand

`ICommand CustomCalibrationLibrary.Views.CalibrationFailed.CalibrationAbortCommand` [get]

Command to abort the calibration

#### 7.9.3.2 CalibrationRestartCommand

`ICommand CustomCalibrationLibrary.Views.CalibrationFailed.CalibrationRestartCommand` [get]

Command to restart the calibration

#### 7.9.3.3 Error

`string CustomCalibrationLibrary.Views.CalibrationFailed.Error` [get], [set]

The error message to be updated on the view.

### 7.9.4 Event Documentation

#### 7.9.4.1 PropertyChanged

`PropertyChangedEventHandler? CustomCalibrationLibrary.Views.CalibrationFailed.PropertyChanged`

The property change event to update the view.

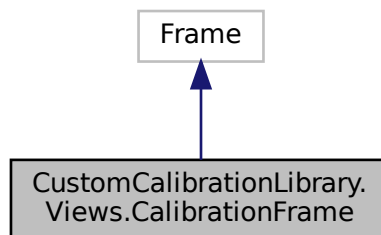
The documentation for this class was generated from the following file:

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

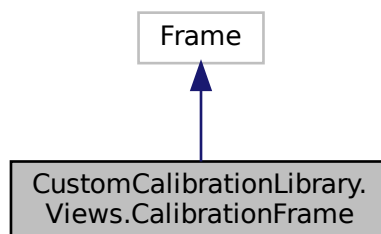
## 7.10 CustomCalibrationLibrary.Views.CalibrationFrame Class Reference

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)

#### 7.10.1 Detailed Description

Interaction logic for CalibrationCollection.xaml

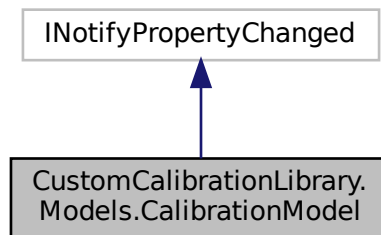
The documentation for this class was generated from the following file:

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

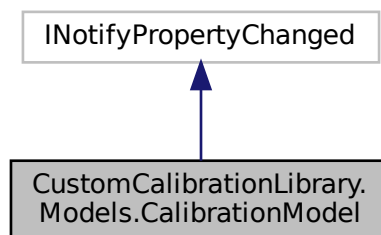
## 7.11 CustomCalibrationLibrary.Models.CalibrationModel Class Reference

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)
- **CalibrationModel** ([TrackerLogger](#) logger, double[][] points)
- void **UpdateGazePoint** (double x, double y)  
*Update the normalized gaze point on the screen.*
- void **InitCalibration** ()  
*Initialise the calibration.*
- 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 **SetCalibrationResult** (List< [GazeCalibrationData](#) > points)  
*Updates the calibration results on the screen.*

## Properties

- string [Error](#) [get, set]  
*The error message of the calibration process.*
- [GazeValidationData](#) **ValidationData** [get, set]
- [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](#)  
*Events to trigger changes in the calibration process.*
- PropertyChangedEventHandler? **PropertyChanged**
- EventHandler< Point >? **GazePointChanged**
- EventHandler< [UserPositionData](#) >? **UserPositionGuideChanged**

### 7.11.1 Detailed Description

The model for the calibration process.

### 7.11.2 Member Function Documentation

#### 7.11.2.1 GazeDataCollected()

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

Trigger the data collected events.

### 7.11.2.2 InitCalibration()

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

Initialise the calibration.

### 7.11.2.3 NextCalibrationPoint()

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

Trigger the next calibration point.

### 7.11.2.4 RedoCalibrationPoint()

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

Remove and re-add the current calibration point

### 7.11.2.5 SetCalibrationResult()

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

Updates the calibration results on the screen.

#### Parameters

<i>points</i>	
---------------	--

### 7.11.2.6 UpdateGazePoint()

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

Update the normalized gaze point on the screen.

#### Parameters

<i>x</i>	The x coordinate
<i>y</i>	The y coordinate



### 7.11.3 Property Documentation

#### 7.11.3.1 CalibrationPoints

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

The calibration points to be added during the calibration process.

#### 7.11.3.2 Error

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

The error message of the calibration process.

#### 7.11.3.3 GazePoint

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

The gaze point position.

#### 7.11.3.4 Index

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

The index of the current calibration point

#### 7.11.3.5 LastStatus

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

The calibration status before an error occurred.

### 7.11.3.6 Points

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

All calibration points.

### 7.11.3.7 Status

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

The status of the calibration process.

### 7.11.3.8 UserPositionGuide

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

The user position guide values.

## 7.11.4 Event Documentation

### 7.11.4.1 CalibrationEvent

```
EventHandler<CalibrationEventType>? CustomCalibrationLibrary.Models.CalibrationModel.Calibration↔  
Event
```

Events to trigger changes in the calibration process.

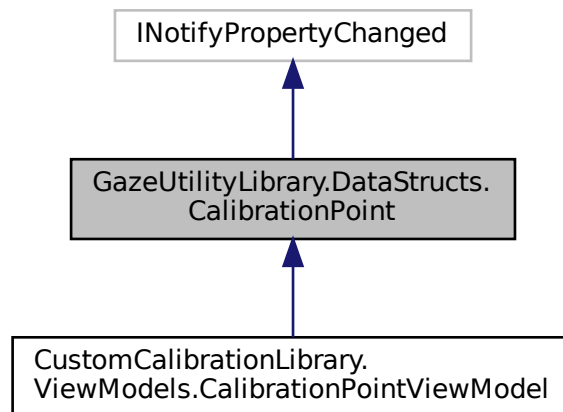
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Models/CalibrationModel.cs

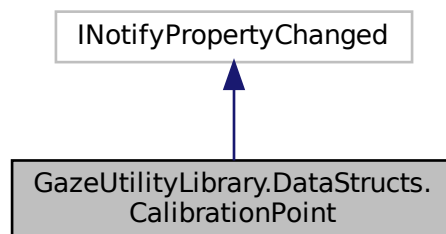
## 7.12 GazeUtilityLibrary.DataStructs.CalibrationPoint Class Reference

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)

### 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.*
- 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 [GazePositionLeft](#) [get, set]  
*The left gaze point.*
- Point [GazePositionRight](#) [get, set]  
*The right gaze point.*

## Events

- PropertyChangedEventHandler? **PropertyChanged**

### 7.12.1 Detailed Description

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

### 7.12.2 Property Documentation

#### 7.12.2.1 GazePositionAverage

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

The average between the left and the right gaze point.

#### 7.12.2.2 GazePositionLeft

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

The left gaze point.

#### 7.12.2.3 GazePositionRight

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

The right gaze point.

#### 7.12.2.4 HasData

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

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

#### 7.12.2.5 Index

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

The index of the calibration point.

#### 7.12.2.6 Position

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

The position of the calibration point.

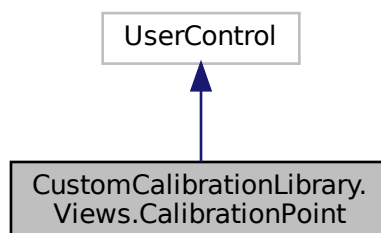
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/CalibrationPoint.cs

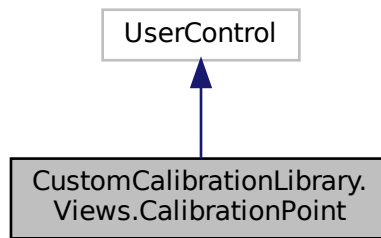
## 7.13 CustomCalibrationLibrary.Views.CalibrationPoint Class Reference

Interaction logic for CalibrationPoint.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationPoint:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationPoint:



### 7.13.1 Detailed Description

Interaction logic for CalibrationPoint.xaml

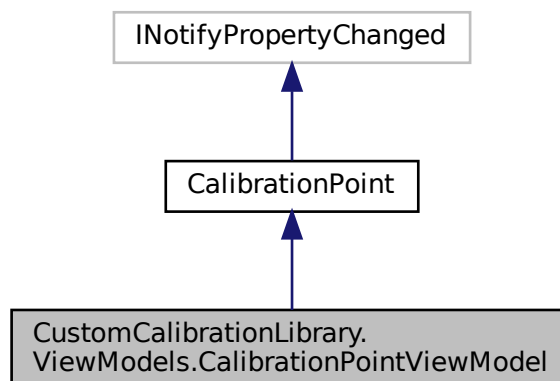
The documentation for this class was generated from the following file:

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

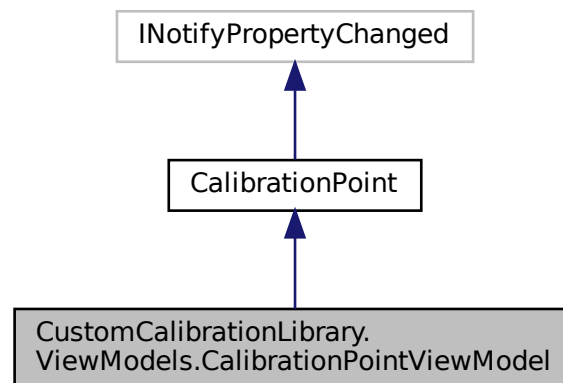
## 7.14 CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel Class Reference

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)
- **CalibrationPointViewModel** ([CalibrationPoint](#) model)

## Additional Inherited Members

### 7.14.1 Detailed Description

The view model for a calibration point.

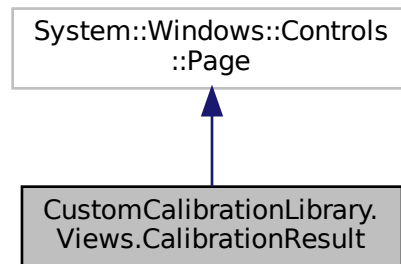
The documentation for this class was generated from the following file:

- `source/CustomCalibrationLibrary/ViewModels/CalibrationPointViewModel.cs`

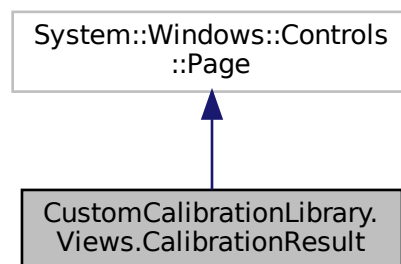
## 7.15 CustomCalibrationLibrary.Views.CalibrationResult Class Reference

Interaction logic for CalibrationResult.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationResult:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationResult:



## Public Member Functions

- **CalibrationResult** ([CalibrationModel](#) model)

### 7.15.1 Detailed Description

Interaction logic for CalibrationResult.xaml

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

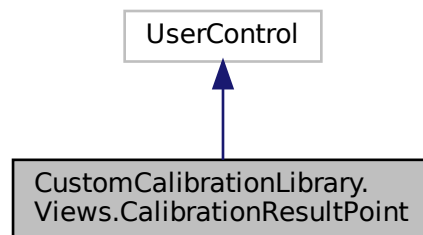
- `source/CustomCalibrationLibrary/Views/CalibrationResult.xaml.cs`



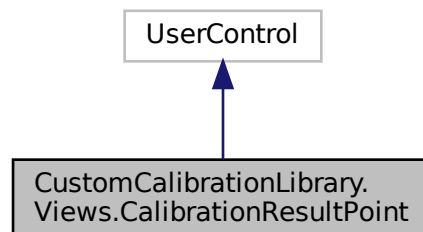
## 7.16 CustomCalibrationLibrary.Views.CalibrationResultPoint Class Reference

Interaction logic for CalibrationResultPoint.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationResultPoint:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationResultPoint:



### 7.16.1 Detailed Description

Interaction logic for CalibrationResultPoint.xaml

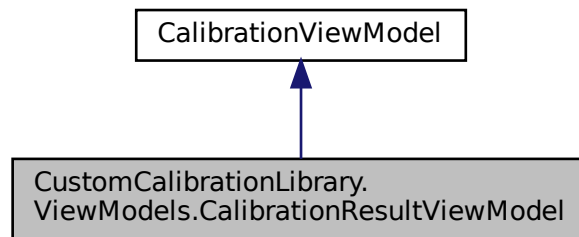
The documentation for this class was generated from the following file:

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

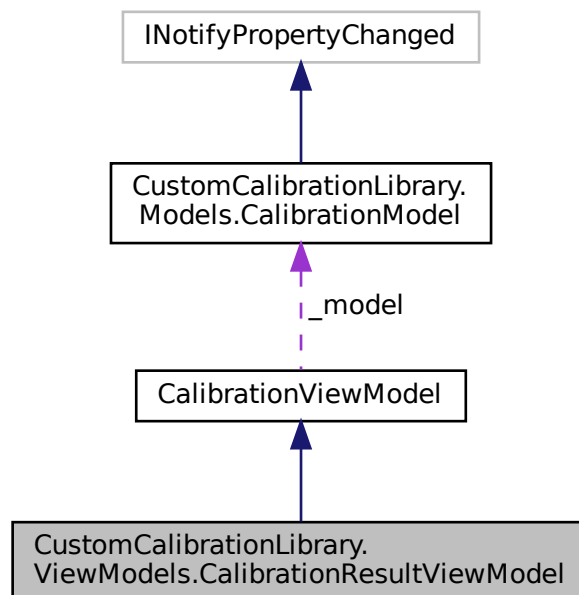
## 7.17 CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel Class Reference

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*

## Additional Inherited Members

### 7.17.1 Detailed Description

View model class of the gaze calibration result.

### 7.17.2 Constructor & Destructor Documentation

#### 7.17.2.1 CalibrationResultViewModel()

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

Constructor

Parameters

<i>model</i>	The claibration model
--------------	-----------------------

### 7.17.3 Member Function Documentation

#### 7.17.3.1 OnGazeToggle()

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

Toggle the visibility of the live gaze point.

## 7.17.4 Property Documentation

### 7.17.4.1 CalibrationAcceptCommand

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

Command to accept the calibration

### 7.17.4.2 CalibrationRestartCommand

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

Command to restart the calibration

### 7.17.4.3 GazePoint

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

The position of the live gaze point

### 7.17.4.4 GazeVisibilityCommand

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

Command to toggle the visibility of the live gaze point

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

- source/CustomCalibrationLibrary/ViewModels/CalibrationResultViewModel.cs

## 7.18 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** ()

## 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.*

### 7.18.1 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.

### 7.18.2 Property Documentation

#### 7.18.2.1 AccuracyLeftEye

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

The accuracy in degrees for the left eye.

#### 7.18.2.2 AccuracyRightEye

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

The accuracy in degrees for the right eye.

### 7.18.2.3 Coordinates

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

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

### 7.18.2.4 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.

### 7.18.2.5 PrecisionLeftEye

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

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

### 7.18.2.6 PrecisionRightEye

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

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

### 7.18.2.7 PrecisionRMSLeftEye

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

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

### 7.18.2.8 PrecisionRMSRightEye

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

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

### 7.18.2.9 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

## 7.19 Tobii.Research.Addons.CalibrationValidationResult Class Reference

Contains the result of the calibration validation.

### Public Member Functions

- override string **ToString** ()

### 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.*

### 7.19.1 Detailed Description

Contains the result of the calibration validation.

### 7.19.2 Property Documentation

### 7.19.2.1 AverageAccuracyLeftEye

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

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

### 7.19.2.2 AverageAccuracyRightEye

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

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

### 7.19.2.3 AveragePrecisionLeftEye

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

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

### 7.19.2.4 AveragePrecisionRightEye

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

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

### 7.19.2.5 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.

### 7.19.2.6 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.



### 7.19.2.7 Points

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

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

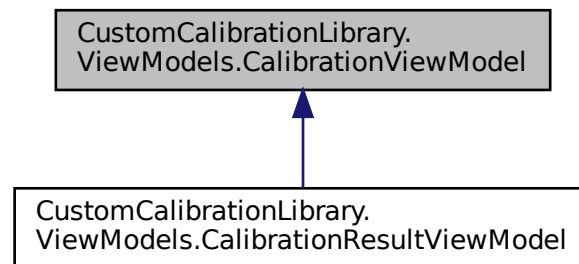
The documentation for this class was generated from the following file:

- source/TobiiProSdkAddons/ScreenBasedCalibrationValidation.cs

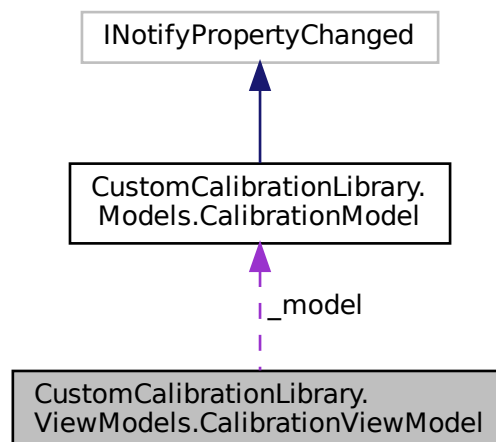
## 7.20 CustomCalibrationLibrary.ViewModels.CalibrationViewModel Class Reference

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

## Properties

- [ObservableCollection](#)< [CalibrationPointViewModel](#) > [CalibrationPoints](#) [get]  
*The collection of calibration points to be shown on the view*

### 7.20.1 Detailed Description

The view model class of the calibration view

### 7.20.2 Constructor & Destructor Documentation

#### 7.20.2.1 CalibrationViewModel()

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

Constructor

Parameters

<i>model</i>	The calibration model
--------------	-----------------------

### 7.20.3 Property Documentation

#### 7.20.3.1 CalibrationPoints

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

The collection of calibration points to be shown on the view

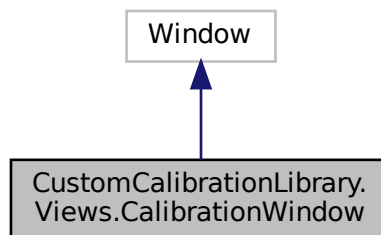
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/CalibrationViewModel.cs

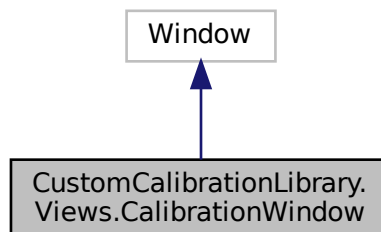
## 7.21 CustomCalibrationLibrary.Views.CalibrationWindow Class Reference

Interaction logic for MainWindow.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationWindow:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationWindow:



### 7.21.1 Detailed Description

Interaction logic for MainWindow.xaml

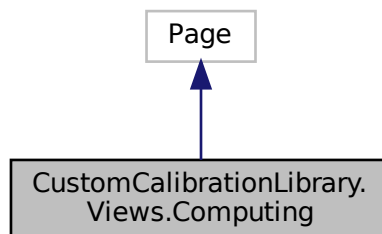
The documentation for this class was generated from the following file:

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

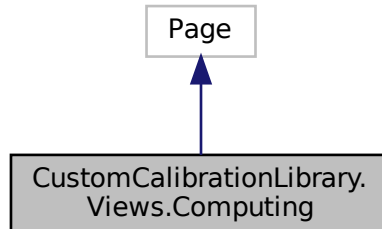
## 7.22 CustomCalibrationLibrary.Views.Computing Class Reference

Interaction logic for Computing.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.Computing:



Collaboration diagram for CustomCalibrationLibrary.Views.Computing:



### 7.22.1 Detailed Description

Interaction logic for Computing.xaml

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

- `source/CustomCalibrationLibrary/Views/Computing.xaml.cs`

## 7.23 GazeUtilityLibrary.ConfigItem Class Reference

configuration file class

## Properties

- string? **ConfigName** [get, set]
- string **DataLogColumnOrder** [get, set]
- string[] **DataLogColumnTitle** [get, set]
- string **CalibrationLogColumnOrder** [get, set]
- string[] **CalibrationLogColumnTitle** [get, set]
- string **ValidationLogColumnOrder** [get, set]
- string[] **ValidationLogColumnTitle** [get, set]
- int **DataLogCount** [get, set]
- string **DataLogFormatDiameter** [get, set]
- string **DataLogFormatOrigin** [get, set]
- string **DataLogFormatNormalizedPoint** [get, set]
- string **DataLogFormatTimeStamp** [get, set]
- string **DataLogFormatTimeStampRelative** [get, set]
- string **DataLogPath** [get, set]
- bool **DataLogWriteOutput** [get, set]
- bool **CalibrationLogWriteOutput** [get, set]
- bool **ValidationLogWriteOutput** [get, set]
- double[][] **CalibrationPoints** [get, set]
- double[][] **ValidationPoints** [get, set]
- bool **DataLogDisabledOnStartup** [get, set]
- double **DispersionThreshold** [get, set]
- double **DriftCompensationTimer** [get, set]
- string? **LicensePath** [get, set]
- bool **MouseControl** [get, set]
- bool **MouseControlHide** [get, set]
- bool **MouseCalibrationHide** [get, set]
- string **MouseStandardIconPath** [get, set]
- int **ReadyTimer** [get, set]
- int **TrackerDevice** [get, set]
- string **TobiiApplicationPath** [get, set]
- string **TobiiCalibrate** [get, set]
- string **TobiiCalibrateArguments** [get, set]
- [ConfigScreenArea](#) **ScreenArea** [get, set]

### 7.23.1 Detailed Description

configuration file class

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

- [source/GazeUtilityLibrary/GazeConfiguration.cs](#)

## 7.24 GazeUtilityLibrary.ConfigScreenArea Class Reference

### Public Member Functions

- **ConfigScreenArea** ([ScreenArea](#) screenArea)

## Properties

- double **Width** [get, set]
- double **Height** [get, set]
- double[] **Center** [get, set]
- double[] **TopLeft** [get, set]
- double[] **TopRight** [get, set]
- double[] **BottomLeft** [get, set]
- double[] **BottomRight** [get, set]

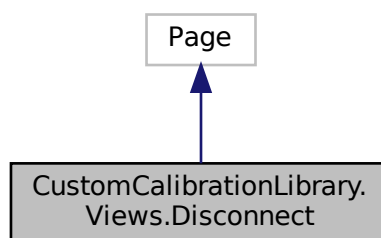
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeConfiguration.cs

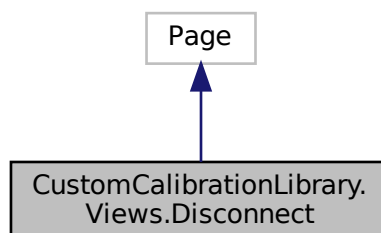
## 7.25 CustomCalibrationLibrary.Views.Disconnect Class Reference

Interaction logic for Disconnect.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.Disconnect:



Collaboration diagram for CustomCalibrationLibrary.Views.Disconnect:



## Public Member Functions

- **Disconnect** ([CalibrationModel](#) model)

## Properties

- ICommand [CalibrationAbortCommand](#) [get]  
*Command to abort the calibration*

### 7.25.1 Detailed Description

Interaction logic for Disconnect.xaml

### 7.25.2 Property Documentation

#### 7.25.2.1 CalibrationAbortCommand

ICommand CustomCalibrationLibrary.Views.Disconnect.CalibrationAbortCommand [get]

Command to abort the calibration

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

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

## 7.26 GazeUtilityLibrary.DriftCompensation Class Reference

The class to handle drift compensation.

## Public Member Functions

- **DriftCompensation** (Vector3 fixationPoint, int fixationFrameCount, double dispersionThreshold)
- 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.*

### 7.26.1 Detailed Description

The class to handle drift compensation.

### 7.26.2 Member Function Documentation

#### 7.26.2.1 Reset()

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

Reset the drift compensation quaternion to the identity.

#### 7.26.2.2 Start()

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

Start the drift compensation.

#### 7.26.2.3 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

<i>gazeData</i>	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.



### 7.26.3 Property Documentation

#### 7.26.3.1 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

## 7.27 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*

### 7.27.1 Detailed Description

The drift compensation data structure

### 7.27.2 Constructor & Destructor Documentation

#### 7.27.2.1 DriftCompensationData()

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

Constructor

## Parameters

<i>screen</i>	The screen area
<i>driftCompensation</i>	The drift compensation quaternion
<i>gazeData</i>	The 3d gaze data structure

## 7.27.3 Property Documentation

### 7.27.3.1 Compensation

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

The drift compensation quaternion

### 7.27.3.2 GazePosition2d

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

The drift compensated 2d gaze position

### 7.27.3.3 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`

## 7.28 CustomCalibrationLibrary.ViewModels.DriftCompensationView↔ Model Class Reference

The view model class of the drift compensation view.

## Public Member Functions

- [DriftCompensationViewModel](#) ()

*Constructor*

## Properties

- [CalibrationPoint](#) [FixationPoint](#) [get, set]

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

### 7.28.1 Detailed Description

The view model class of the drift compensation view.

### 7.28.2 Constructor & Destructor Documentation

#### 7.28.2.1 DriftCompensationViewModel()

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

Constructor

### 7.28.3 Property Documentation

#### 7.28.3.1 FixationPoint

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

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

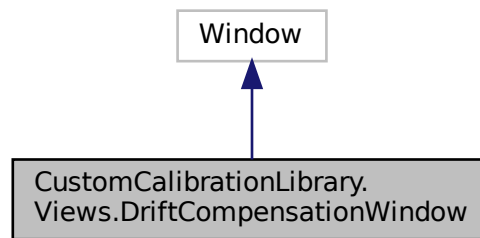
The documentation for this class was generated from the following file:

- `source/CustomCalibrationLibrary/ViewModels/DriftCompensationViewModel.cs`

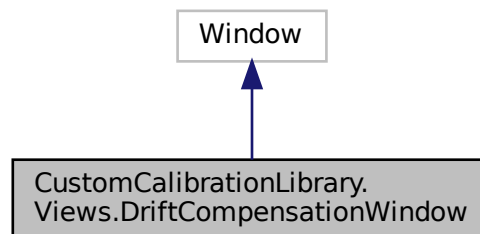
## 7.29 CustomCalibrationLibrary.Views.DriftCompensationWindow Class Reference

Interaction logic for DriftCompensation.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.DriftCompensationWindow:



Collaboration diagram for CustomCalibrationLibrary.Views.DriftCompensationWindow:



### 7.29.1 Detailed Description

Interaction logic for DriftCompensation.xaml

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

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

## 7.30 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*

### 7.30.1 Detailed Description

The eye data set, including pupil information.

### 7.30.2 Constructor & Destructor Documentation

#### 7.30.2.1 EyeData()

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

Initializes a new instance of the [EyeData](#) class.

#### Parameters

<i>pupilDiameter</i>	The pupil diameter.
<i>isPupilDiameterValid</i>	The validity of the pupil diameter.

### 7.30.3 Property Documentation

#### 7.30.3.1 IsPupilDiameterValid

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

The validity flag of th epupil diameter

### 7.30.3.2 PupilDiameter

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

The diameter of the pupil

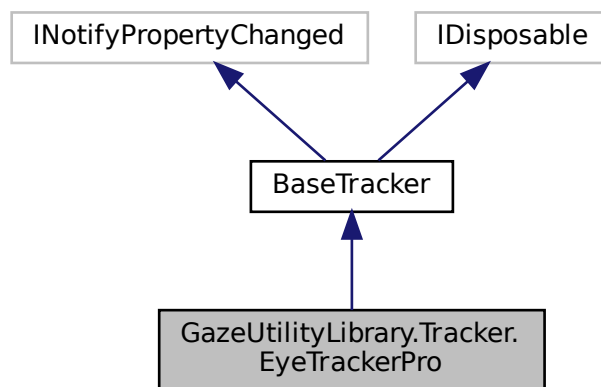
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/EyeData.cs

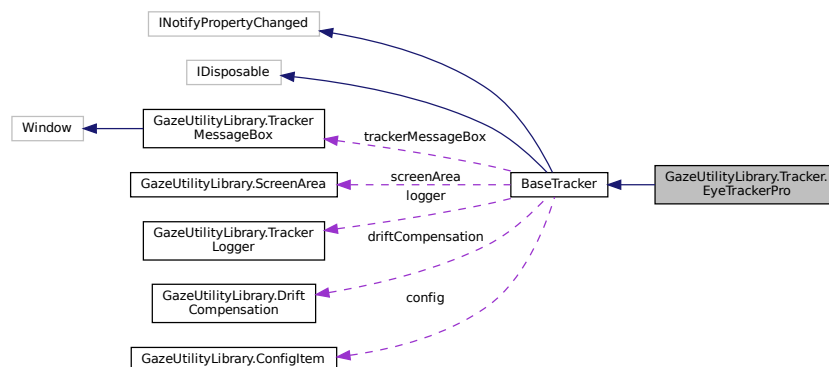
## 7.31 GazeUtilityLibrary.Tracker.EyeTrackerPro Class Reference

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 Tobii 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](#) ()  
*Get the number of required gaze samples to compute a fixation.*
- override Vector3 [GetUnitDirection](#) ()  
*Get the unit vector pointing in the direction of the gaze vector.*

## Additional Inherited Members

### 7.31.1 Detailed Description

Interface to the [Tobii](#) SDK Pro engine

See also

[GazeHelper.TrackerHandler](#)

## 7.31.2 Constructor & Destructor Documentation

### 7.31.2.1 EyeTrackerPro()

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

Initializes a new instance of the [EyeTrackerPro](#) class.

#### Parameters

<i>logger</i>	The logger.
<i>ready_timer</i>	The ready timer.
<i>license_path</i>	The license path.

## 7.31.3 Member Function Documentation

### 7.31.3.1 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](#).

### 7.31.3.2 CollectCalibrationDataAsync()

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

Collects gaze data of a calibration point.



**Parameters**

<i>point</i>	
--------------	--

**Returns**

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

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

**7.31.3.3 CollectValidationDataAsync()**

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

Collects gaze data of a validation point.

**Parameters**

<i>point</i>	
--------------	--

**Returns**

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

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

**7.31.3.4 ComputeValidation()**

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

Compute the validation data.

**Returns**

The validation data result.

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.31.3.5 FinishCalibration()

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

Finish the screen based calibration process.

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.31.3.6 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](#).

### 7.31.3.7 FinishValidation()

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

Finish the screen based validation process.

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.31.3.8 GetFixationFrameCount()

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

Get the number of required gaze samples to compute a fixation.

#### Returns

60

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.31.3.9 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](#).

### 7.31.3.10 InitCalibration()

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

Initialise the screen based calibration.

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.31.3.11 InitCalibrationAsync()

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

Initialise the screen based calibration.

#### Returns

An async handler

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.31.3.12 InitDriftCompensation()

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

Initialise the drift compensation.

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.31.3.13 InitValidation()

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

Initialise the screen based calibration.

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.31.3.14 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](#).

### 7.31.3.15 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.

### 7.31.3.16 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](#).

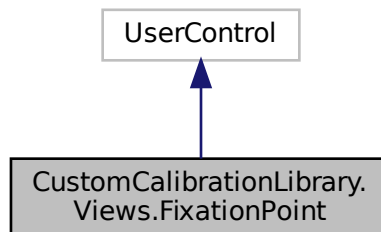
The documentation for this class was generated from the following file:

- [source/GazeUtilityLibrary/Tracker/EyeTrackerPro.cs](#)

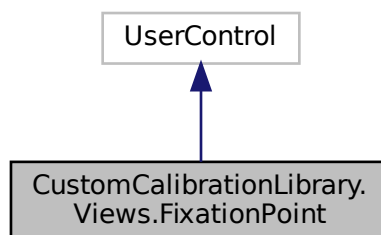
## 7.32 CustomCalibrationLibrary.Views.FixationPoint Class Reference

Interaction logic for FixationPoint.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.FixationPoint:



Collaboration diagram for CustomCalibrationLibrary.Views.FixationPoint:



### 7.32.1 Detailed Description

Interaction logic for FixationPoint.xaml

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

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

## 7.33 GazeUtilityLibrary.DataStructs.GazeCalibrationData Class Reference

The event argument class for [Tobii](#) eyetracker data

## Public Member Functions

- [GazeCalibrationData](#) (double xCoord, double yCoord, double xCoordLeft, double yCoordLeft, bool validityLeft, double xCoordRight, double yCoordRight, bool validityRight)  
*Initializes a new instance of the GazeDataArgs class.*
- string[] [Prepare](#) ([ConfigItem](#) config)  
*Prepare a list of formatted calibration data values*

## Properties

- double **XCoord** [get]
- double **YCoord** [get]
- double **XCoordLeft** [get]
- double **YCoordLeft** [get]
- bool **ValidityLeft** [get]
- double **XCoordRight** [get]
- double **YCoordRight** [get]
- bool **ValidityRight** [get]

### 7.33.1 Detailed Description

The event argument class for [Tobii](#) eyetracker data

### 7.33.2 Constructor & Destructor Documentation

#### 7.33.2.1 GazeCalibrationData()

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

Initializes a new instance of the GazeDataArgs class.

#### Parameters

<i>xCoord</i>	The x coord of the calibration point.
<i>yCoord</i>	The y coord of the calibration point.
<i>xCoordLeft</i>	The x coord of the gaze point of the left eye.
<i>yCoordLeft</i>	The y coord of the gaze point of the left eye.
<i>validityLeft</i>	the validity of gaze point coordinate of the left eye.
<i>xCoordRight</i>	The x coord of the gaze point of the right eye.
<i>yCoordRight</i>	The y coord of the gaze point of the right eye.
<i>validityRight</i>	the validity of gaze point coordinate of the right eye.

### 7.33.3 Member Function Documentation

#### 7.33.3.1 Prepare()

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

Prepare a list of formatted calibration data values

##### Parameters

<i>config</i>	The gaze configuration structure
---------------	----------------------------------

##### Returns

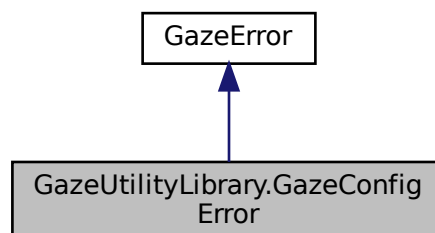
A list of formatted values. Each index corresponds to a specific value. This allows to reorder the list according to a format string.

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

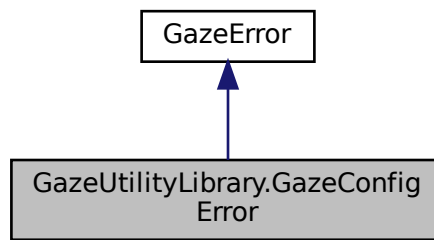
- source/GazeUtilityLibrary/DataStructs/GazeCalibrationData.cs

## 7.34 GazeUtilityLibrary.GazeConfigError Class Reference

Inheritance diagram for GazeUtilityLibrary.GazeConfigError:



Collaboration diagram for GazeUtilityLibrary.GazeConfigError:



## Public Member Functions

- string [GetGazeConfigErrorString](#) ()  
*Gets the gaze error string.*

## Properties

- [EGazeConfigError](#) **Error** [set]

## Additional Inherited Members

### 7.34.1 Member Function Documentation

#### 7.34.1.1 GetGazeConfigErrorString()

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

Gets the gaze error string.

#### Returns

the error string with binary error values if errors occurred, the empty string otherwise

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

- `source/GazeUtilityLibrary/GazeError.cs`



## 7.35 GazeUtilityLibrary.GazeConfiguration Class Reference

### Public Member Functions

- **GazeConfiguration** ([TrackerLogger](#) logger)
- 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]

#### 7.35.1 Member Function Documentation

##### 7.35.1.1 CleanupCalibrationOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.CleanupCalibrationOutputFile (
    string error ) [inline]
```

Close the calibration outputfile and rename it by appending error codes.

#### Parameters

<i>error</i>	
--------------	--

**Returns**

True on success, False on failure.

**7.35.1.2 CleanupGazeOutputFile()**

```
bool GazeUtilityLibrary.GazeConfiguration.CleanupGazeOutputFile (
    string error ) [inline]
```

Close the gaze outputfile and rename it by appending error codes.

**Parameters**

<i>error</i>	
--------------	--

**Returns**

True on success, False on failure.

**7.35.1.3 CleanupValidationOutputFile()**

```
bool GazeUtilityLibrary.GazeConfiguration.CleanupValidationOutputFile (
    string error ) [inline]
```

Close the validation outputfile and rename it by appending error codes.

**Parameters**

<i>error</i>	
--------------	--

**Returns**

True on success, False on failure.

**7.35.1.4 DumpCurrentConfigurationFile()**

```
bool GazeUtilityLibrary.GazeConfiguration.DumpCurrentConfigurationFile ( ) [inline]
```

Dump current configuration to the disk.

**Returns**

True on success, False on failure.

### 7.35.1.5 InitConfig()

```
bool GazeUtilityLibrary.GazeConfiguration.InitConfig ( ) [inline]
```

Initialise the gaze configuration by parsing and checking the configuration file.

#### Returns

True on success, False on failure.

### 7.35.1.6 PrepareCalibrationOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.PrepareCalibrationOutputFile (
    string? subjectCode ) [inline]
```

Prepare the calibration output file based on the configuration.

#### Parameters

<i>subjectCode</i>	An optional subject code to be appended to the file name if set.
--------------------	--

#### Returns

True on success, False on failure.

### 7.35.1.7 PrepareGazeOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.PrepareGazeOutputFile (
    string? subjectCode,
    string? outputPath ) [inline]
```

Prepare the gaze output file based on the configuration.

#### Parameters

<i>subjectCode</i>	An optional subject code to be appended to the file name if set.
<i>outputPath</i>	An optional output path where the file will be stored.

#### Returns

True on success, False on failure.

### 7.35.1.8 PrepareValidationOutputFile()

```
bool GazeUtilityLibrary.GazeConfiguration.PrepareValidationOutputFile (
    string? subjectCode ) [inline]
```

Prepare the validation output file based on the configuration.

#### Parameters

<i>subjectCode</i>	An optional subject code to be appended to the file name if set.
--------------------	--

#### Returns

True on success, False on failure.

### 7.35.1.9 WriteToCalibrationOutput()

```
void GazeUtilityLibrary.GazeConfiguration.WriteToCalibrationOutput (
    string[] formatted_values ) [inline]
```

Write to the calibration output file

#### Parameters

<i>formatted_values</i>	The list of formatted values to be written to the file.
-------------------------	---

### 7.35.1.10 WriteToGazeOutput()

```
void GazeUtilityLibrary.GazeConfiguration.WriteToGazeOutput (
    string[] formatted_values ) [inline]
```

Write to the gaze output file

#### Parameters

<i>formatted_values</i>	The list of formatted values to be written to the file.
-------------------------	---

### 7.35.1.11 WriteToValidationOutput()

```
void GazeUtilityLibrary.GazeConfiguration.WriteToValidationOutput (
    string[] formatted_values ) [inline]
```

Write to the calibration output file

## Parameters

<i>formatted_values</i>	The list of formatted values to be written to the file.
-------------------------	---

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

- source/GazeUtilityLibrary/GazeConfiguration.cs

## 7.36 GazeUtilityLibrary.DataStructs.GazeData Class Reference

The class definition of a gaze data set

### Public Member Functions

- [GazeData](#) (TimeSpan timestamp, Vector2 gazePoint2d, bool isGazePoint2dValid)  
*Initializes a new instance of the GazeDataArgs class.*
- [GazeData](#) (TimeSpan timestamp, Vector2 gazePoint2dLeft, bool isGazePoint2dValidLeft, Vector2 gazePoint2dRight, bool isGazePoint2dValidRight)  
*Initializes a new instance of the GazeDataArgs class.*
- [GazeData](#) (TimeSpan timestamp, 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)  
*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, set]
- [GazeDataCollection?](#) **Left** [get]
- [GazeDataCollection?](#) **Right** [get]
- [GazeDataCollection](#) **Combined** [get]
- [DriftCompensationData?](#) **DriftCompensation** [get, set]

### 7.36.1 Detailed Description

The class definition of a gaze data set

### 7.36.2 Constructor & Destructor Documentation

#### 7.36.2.1 GazeData() [1/3]

```
GazeUtilityLibrary.DataStructs.GazeData.GazeData (
    TimeSpan timestamp,
    Vector2 gazePoint2d,
    bool isGazePoint2dValid ) [inline]
```

Initializes a new instance of the GazeDataArgs class.

## Parameters

<i>timestamp</i>	The timestamp.
<i>gazePoint2d</i>	The 2d coordinates of the combined gaze point.
<i>isGazePoint2dValid</i>	The validity of the combined 2d gaze point.

**7.36.2.2 GazeData() [2/3]**

```
GazeUtilityLibrary.DataStructs.GazeData.GazeData (
    TimeSpan timestamp,
    Vector2 gazePoint2dLeft,
    bool isGazePoint2dValidLeft,
    Vector2 gazePoint2dRight,
    bool isGazePoint2dValidRight ) [inline]
```

Initializes a new instance of the GazeDataArgs class.

## Parameters

<i>timestamp</i>	The timestamp.
<i>gazePoint2dLeft</i>	The 2d coordinates of the left gaze point.
<i>isGazePoint2dValidLeft</i>	The validity of the left 2d gaze point.
<i>gazePoint2dRight</i>	The 2d coordinates of the right gaze point.
<i>isGazePoint2dValidRight</i>	The validity of the right 2d gaze point.

**7.36.2.3 GazeData() [3/3]**

```
GazeUtilityLibrary.DataStructs.GazeData.GazeData (
    TimeSpan timestamp,
    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

<i>timestamp</i>	The timestamp.
<i>gazePoint2dLeft</i>	The 2d coordinates of the left gaze point.
<i>isGazePoint2dValidLeft</i>	The validity of the left 2d gaze point.
<i>gazePoint2dRight</i>	The 2d coordinates of the right gaze point.
<i>isGazePoint2dValidRight</i>	The validity of the right 2d gaze point.
<i>gazePoint3dLeft</i>	The 3d coordinates of the left gaze point.
<i>isGazePoint3dValidLeft</i>	The validity of the left 3d gaze point.
<i>gazePoint3dRight</i>	The 3d coordinates of the right gaze point.
<i>isGazePoint3dValidRight</i>	The validity of the right 3d gaze point.
<i>gazeOrigin3dLeft</i>	The 3d coordinates of the left gaze origin.
<i>isGazeOrigin3dValidLeft</i>	The validity of the left 3d gaze origin.
<i>gazeOrigin3dRight</i>	The 3d coordinates of the right gaze origin.
<i>isGazeOrigin3dValidRight</i>	The validity of the right 3d gaze origin.
<i>pupilDiameterLeft</i>	The pupil diameter the left eye.
<i>isPupilDiameterValidLeft</i>	The validity of the left pupil diameter.
<i>pupilDiameterRight</i>	The pupil diameter the left eye.
<i>isPupilDiameterValidRight</i>	The validity of the left pupil diameter.

### 7.36.3 Member Function Documentation

#### 7.36.3.1 Prepare()

```
string [] GazeUtilityLibrary.DataStructs.GazeData.Prepare (
    ConfigItem config,
    int trialId,
    string tag,
    TimeSpan startTime ) [inline]
```

Prepare a list of formatted gaze data values

## Parameters

<i>config</i>	The gaze configuration structure
<i>trialId</i>	The ID of the current trial.
<i>tag</i>	An arbitrary tag to associate with the data sample.
<i>startTime</i>	The system time to use to 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.

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

- source/GazeUtilityLibrary/DataStructs/GazeData.cs

## 7.37 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]
- bool **IsGazePointValid** [get]

#### 7.37.1 Detailed Description

The 2d gaze data set.

#### 7.37.2 Constructor & Destructor Documentation

##### 7.37.2.1 GazeData2d()

```
GazeUtilityLibrary.DataStructs.GazeData2d.GazeData2d (  
    Vector2 gazePoint,  
    bool isGazePointValid ) [inline]
```

Initializes a new instance of the [GazeData2d](#) class.

##### Parameters

<i>gazePoint</i>	The 2d coordinates of the gaze point.
<i>isGazePointValid</i>	The validity of the 2d gaze point.

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

- source/GazeUtilityLibrary/DataStructs/GazeData2d.cs

## 7.38 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]
- bool **IsGazePointValid** [get]
- Vector3 **GazeOrigin** [get]
- Vector3 **GazeDirection** [get]
- float **GazeDistance** [get]
- bool **IsGazeOriginValid** [get]

### 7.38.1 Detailed Description

The 3d gaze data set.

### 7.38.2 Constructor & Destructor Documentation

#### 7.38.2.1 GazeData3d()

```
GazeUtilityLibrary.DataStructs.GazeData3d.GazeData3d (
    Vector3 gazePoint,
    bool isGazePointValid,
    Vector3 gazeOrigin,
    bool isGazeOriginValid ) [inline]
```

Initializes a new instance of the [GazeData3d](#) class.

#### Parameters

<i>gazePoint</i>	The 3d coordinates of the gaze point.
<i>isGazePointValid</i>	The validity of the 3d gaze point.
<i>gazeOrigin</i>	The 3d coordinates of the gaze origin.
<i>isGazeOriginValid</i>	The validity of the 3d gaze origin.

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

- source/GazeUtilityLibrary/DataStructs/GazeData3d.cs

## 7.39 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 isGazePoint3dValid, Vector3 gazeOrigin3d, bool isGazeOrigin3dValid, float pupilDiameter, bool isPupilDiameterValid)  
*Initializes a new instance of the GazeDataItem class.*

## Properties

- [GazeData2d](#) **GazeData2d** [get]
- [GazeData3d?](#) **GazeData3d** [get]
- [EyeData?](#) **EyeData** [get]

### 7.39.1 Detailed Description

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

### 7.39.2 Constructor & Destructor Documentation

#### 7.39.2.1 GazeDataCollection() [1/2]

```
GazeUtilityLibrary.DataStructs.GazeDataCollection.GazeDataCollection (
    Vector2 gazePoint2d,
    bool isGazePoint2dValid ) [inline]
```

Initializes a new instance of the GazeDataItem class.

#### Parameters

<i>gazePoint2d</i>	The 2d coordinates of the gaze point.
<i>isGazePoint2dValid</i>	The validity of the 2d gaze point.

#### 7.39.2.2 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

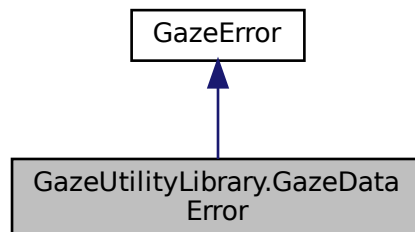
<i>gazePoint2d</i>	The 2d coordinates of the gaze point.
<i>isGazePoint2dValid</i>	The validity of the 2d gaze point.
<i>gazePoint3d</i>	The 3d coordinates of the gaze point.
<i>isGazePoint3dValid</i>	The validity of the 3d gaze point.
<i>gazeOrigin3d</i>	The 3d coordinates of the gaze origin.
<i>isGazeOrigin3dValid</i>	The validity of the 3d gaze origin.
<i>pupilDiameter</i>	The pupil diameter.
<i>isPupilDiameterValid</i>	The validity of the pupil diameter.

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

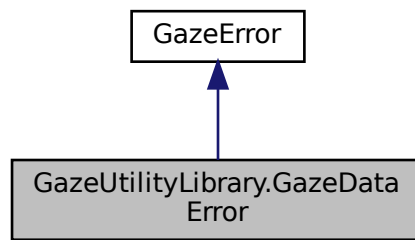
- source/GazeUtilityLibrary/DataStructs/GazeDataCollection.cs

## 7.40 GazeUtilityLibrary.GazeDataError Class Reference

Inheritance diagram for GazeUtilityLibrary.GazeDataError:



Collaboration diagram for GazeUtilityLibrary.GazeDataError:



## Public Member Functions

- string [GetGazeDataErrorString](#) ()  
*Gets the gaze error string.*

## Properties

- [EGazeDataError](#) **Error** [set]

## Additional Inherited Members

### 7.40.1 Member Function Documentation

#### 7.40.1.1 GetGazeDataErrorString()

```
string GazeUtilityLibrary.GazeDataError.GetGazeDataErrorString ( ) [inline]
```

Gets the gaze error string.

#### Returns

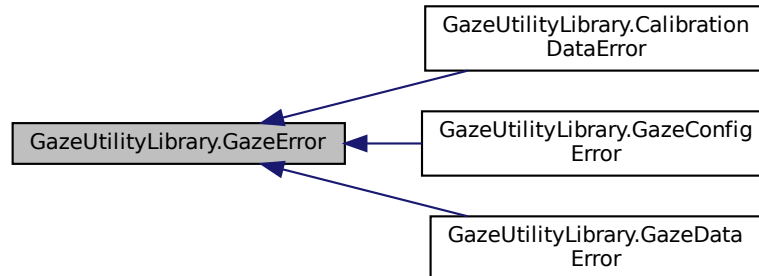
the error string with binary error values if errors occurred, the empty string otherwise

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

- `source/GazeUtilityLibrary/GazeError.cs`

## 7.41 GazeUtilityLibrary.GazeError Class Reference

Inheritance diagram for GazeUtilityLibrary.GazeError:



### Protected Member Functions

- string [ConvertToBinString](#) (int val, int len)  
*Converts a integer value to a binary string.*

### 7.41.1 Member Function Documentation

#### 7.41.1.1 ConvertToBinString()

```
string GazeUtilityLibrary.GazeError.ConvertToBinString (
    int val,
    int len ) [inline], [protected]
```

Converts a integer value to a binary string.

#### Parameters

<i>val</i>	The value.
<i>len</i>	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`

## 7.42 GazeUtilityLibrary.DataStructs.GazeValidationData Class Reference

### Public Member Functions

- **GazeValidationData** (float accuracyLeft, float accuracyRight, float precisionLeft, float precisionRight, float precisionRmsLeft, float precisionRmsRight)
- string[] **Prepare** ([ConfigItem](#) config)  
*Prepare a list of formatted calibration data values*

### Properties

- float **AccuracyLeft** [get]
- float **AccuracyRight** [get]
- float **PrecisionLeft** [get]
- float **PrecisionRight** [get]
- float **PrecisionRmsLeft** [get]
- float **PrecisionRmsRight** [get]

### 7.42.1 Member Function Documentation

#### 7.42.1.1 Prepare()

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

Prepare a list of formatted calibration data values

#### Parameters

<i>config</i>	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.

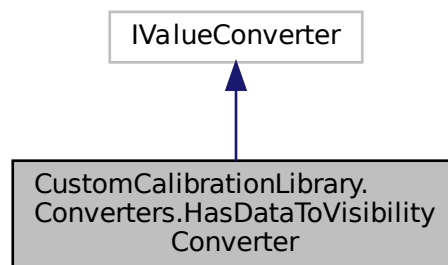
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/GazeValidationData.cs

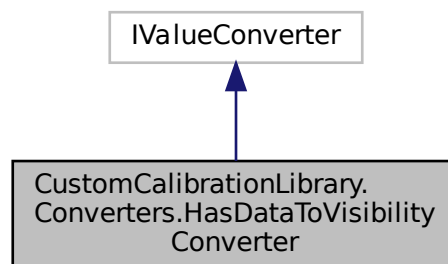
## 7.43 CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter Class Reference

Converts True to Hidden and False to Visible

Inheritance diagram for CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter:



Collaboration diagram for CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter:



## Public Member Functions

- object **Convert** (object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
- object **ConvertBack** (object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)

### 7.43.1 Detailed Description

Converts True to Hidden and False to Visible

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

- source/CustomCalibrationLibrary/Converters/HasDataToVisibilityConverter.cs

## 7.44 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.*

### 7.44.1 Detailed Description

The config file "config.json" is parsed and its values are attributed to the [ConfigItem](#) class.

### 7.44.2 Constructor & Destructor Documentation

#### 7.44.2.1 JsonConfigParser()

```
GazeUtilityLibrary.JsonConfigParser.JsonConfigParser (
    TrackerLogger logger ) [inline]
```

Initializes a new instance of the [JsonConfigParser](#) class.

#### Parameters

<i>logger</i>	The logger.
---------------	-------------

### 7.44.3 Member Function Documentation

#### 7.44.3.1 GetDefaultConfig()

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

Gets the default configuration values.



**Returns**

the default configuration values.

**7.44.3.2 ParseJsonConfig()**

```
ConfigItem? GazeUtilityLibrary.JsonConfigParser.ParseJsonConfig (
    ref GazeConfigError error ) [inline]
```

Parses the json configuration.

**Returns**

the updated [ConfigItem](#) class.

**7.44.3.3 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**

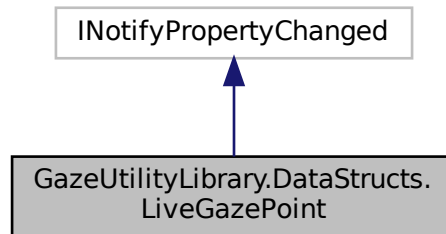
<i>item</i>	The json configuration item.
<i>path</i>	The path where the file will be written.

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

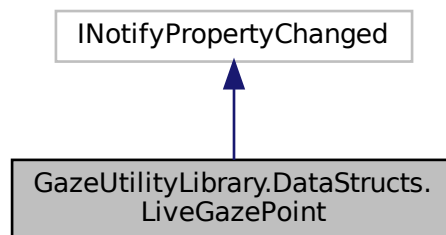
- `source/GazeUtilityLibrary/GazeConfiguration.cs`

## 7.45 GazeUtilityLibrary.DataStructs.LiveGazePoint Class Reference

Inheritance diagram for GazeUtilityLibrary.DataStructs.LiveGazePoint:



Collaboration diagram for GazeUtilityLibrary.DataStructs.LiveGazePoint:



### Properties

- double **X** [get, set]
- double **Y** [get, set]
- bool **Visibility** [get, set]

### Events

- PropertyChangedEventHandler? **PropertyChanged**

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

- `source/GazeUtilityLibrary/DataStructs/LiveGazePoint.cs`

## 7.46 CustomCalibrationLibrary.ViewModels.Monitor Class Reference

### Public Member Functions

- **Monitor** (int index, string name)

### Properties

- string **Name** [get]
- int **Index** [get]

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

- source/CustomCalibrationLibrary/ViewModels/ScreenSelectionViewModel.cs

## 7.47 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.*

### 7.47.1 Detailed Description

hide standard mouse pointer and resore it

### 7.47.2 Constructor & Destructor Documentation

#### 7.47.2.1 MouseHider()

```
GazeUtilityLibrary.MouseHider.MouseHider (
    TrackerLogger logger ) [inline]
```

Initializes a new instance of the [MouseHider](#) class.

**Parameters**

<i>logger</i>	The logger.
---------------	-------------

### 7.47.3 Member Function Documentation

#### 7.47.3.1 HideCursor()

```
void GazeUtilityLibrary.MouseHider.HideCursor ( ) [inline]
```

Hides the cursor.

Hides the standard mouse pointer by replacing the current icon with a transparent icon.

#### 7.47.3.2 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**

<i>pathToCur</i>	The path to the standard mouse pointer icon.
------------------	--

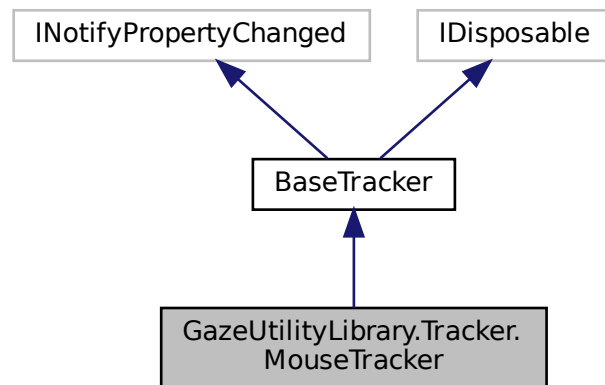
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/MouseHider.cs

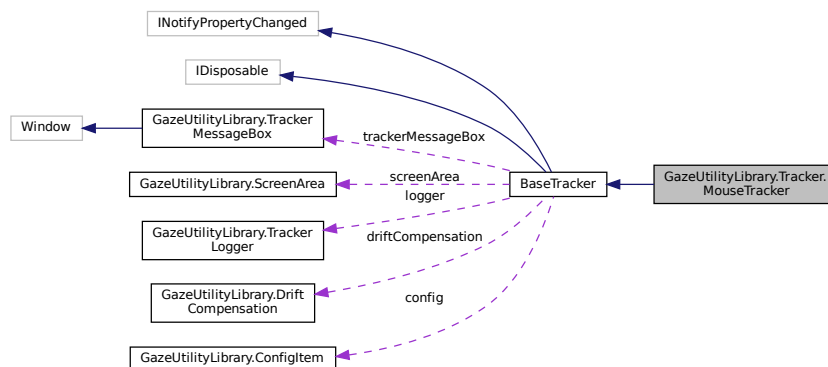
## 7.48 GazeUtilityLibrary.Tracker.MouseTracker Class Reference

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 calibration 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 calibration 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 calibration process. This is device specific and must be overwritten by the device class.*
  - override void [FinishCalibration](#) ()
    - Finish the calibration 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](#) ()
  - 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

### 7.48.1 Detailed Description

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

See also

GazeHelper.TrackerHandler

### 7.48.2 Constructor & Destructor Documentation

#### 7.48.2.1 MouseTracker()

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

Initializes a new instance of the [MouseTracker](#) class.

## Parameters

<i>logger</i>	The logger.
<i>ready_timer</i>	The ready timer.

## 7.48.3 Member Function Documentation

### 7.48.3.1 ApplyCalibration()

```
override Task<List<GazeCalibrationData> > GazeUtilityLibrary.Tracker.MouseTracker.ApplyCalibration ( ) [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](#).

### 7.48.3.2 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

<i>point</i>	The calibration point for which to collect data
--------------	---

## Returns

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

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.48.3.3 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**

<i>point</i>	The calibration point for which to collect data
--------------	---

**Returns**

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

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

**7.48.3.4 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](#).

**7.48.3.5 Dispose()**

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

Releases unmanaged and - optionally - managed resources.

**Parameters**

<i>disposing</i>	true to release both managed and unmanaged resources; false to release only unmanaged resources.
------------------	--

Reimplemented from [GazeUtilityLibrary.Tracker.BaseTracker](#).

**7.48.3.6 FinishCalibration()**

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

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

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).



### 7.48.3.7 FinishCalibrationAsync()

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

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

#### Returns

An async handler

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.48.3.8 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](#).

### 7.48.3.9 GetFixationFrameCount()

```
override int GazeUtilityLibrary.Tracker.MouseTracker.GetFixationFrameCount ( ) [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.

#### Returns

The number of gaze samples to require for fixation detection.

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

### 7.48.3.10 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](#).

#### 7.48.3.11 InitCalibration()

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

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

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

#### 7.48.3.12 InitCalibrationAsync()

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

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

##### Returns

An async handler

Implements [GazeUtilityLibrary.Tracker.BaseTracker](#).

#### 7.48.3.13 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](#).

#### 7.48.3.14 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](#).

#### 7.48.3.15 Start()

```
void GazeUtilityLibrary.Tracker.MouseTracker.Start ( ) [inline]
```

Hooks the callback function HookCallback(int, IntPtr, IntPtr) to mouse events.

### 7.48.3.16 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

## 7.49 GazeUtilityLibrary.DataStructs.PipeCommand Class Reference

### Public Member Functions

- **PipeCommand** (string command, bool reset, string? value)

### Properties

- string **Command** [get, set]
- string? **Value** [get, set]
- bool? **ResetStartTime** [get, set]

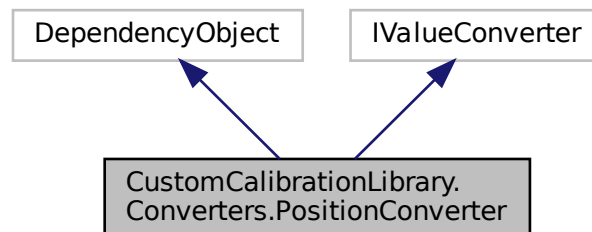
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/PipeCommand.cs

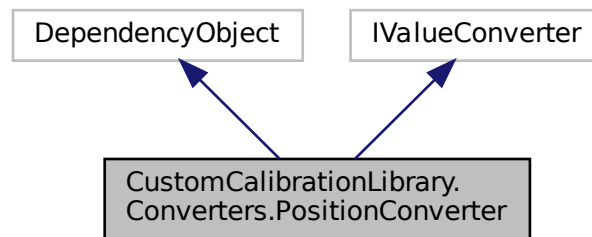
## 7.50 CustomCalibrationLibrary.Converters.PositionConverter Class Reference

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)
- object **ConvertBack** (object value, Type targetType, object parameter, CultureInfo culture)

## Static Public Attributes

- static readonly DependencyProperty **OffsetProperty**

## Properties

- string?? **Offset** [get, set]

### 7.50.1 Detailed Description

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

### 7.50.2 Member Data Documentation

#### 7.50.2.1 OffsetProperty

```
readonly DependencyProperty CustomCalibrationLibrary.Converters.PositionConverter.Offset↔
Property [static]
```

#### Initial value:

```
=
```

```
DependencyProperty.Register("Offset", typeof(string), typeof(PositionConverter), new
PropertyMetadata(null))
```

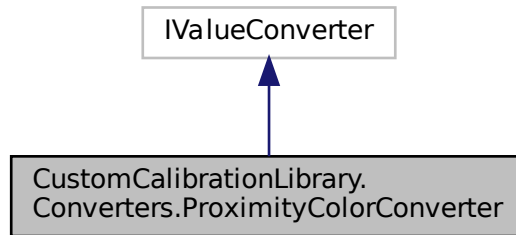
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Converters/PositionConverter.cs

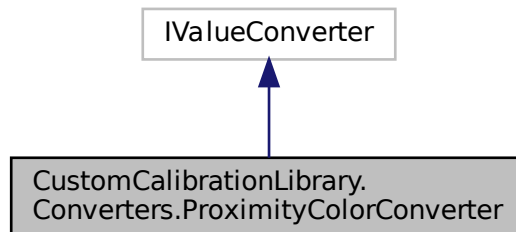
## 7.51 CustomCalibrationLibrary.Converters.ProximityColorConverter Class Reference

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)
- object **ConvertBack** (object value, Type targetType, object parameter, CultureInfo culture)

#### 7.51.1 Detailed Description

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

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

- `source/CustomCalibrationLibrary/Converters/ProximityColorConverter.cs`

## 7.52 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 and 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 screen area.*
- Vector2 [GetPoint2d](#) (Vector3 point)  
*Get the 2d point on the screen given a 3d point on the screen plane.*
- Vector2 [GetPoint2dNormalized](#) (Vector3 point3d)  
*Get the normalized 2d point on the screen given a 3d point on the screen plane. Note that values outside of the interval [0, 1] indicate an intersection point outside of the screen area.*
- bool [Dump](#) (string path, string prefix)  
*Dump the four screen corner points to a csv file*

### Properties

- float **Width** [get]
- float **Height** [get]
- Vector3 **BottomLeft** [get]
- Vector3 **BottomRight** [get]
- Vector3 **TopLeft** [get]
- Vector3 **TopRight** [get]
- Vector3 **Center** [get]

#### 7.52.1 Detailed Description

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

#### 7.52.2 Constructor & Destructor Documentation

##### 7.52.2.1 ScreenArea()

```
GazeUtilityLibrary.ScreenArea.ScreenArea (
    Vector3 bottomLeft,
    Vector3 bottomRight,
    Vector3 topLeft,
    Vector3 topRight,
    float width,
    float height ) [inline]
```

Constructor. Assigns parameters and computes the transformation matrix to transform a 3d point into a 2d point.

## Parameters

<i>bottomLeft</i>	The bottom left 3d coordinate of the screen.
<i>bottomRight</i>	The bottom right 3d coordinate of the screen.
<i>topLeft</i>	The top left 3d coordinate of the screen.
<i>topRight</i>	The top right 3d coordinate of the screen
<i>width</i>	The width of the screen
<i>height</i>	The heigth of the screen

### 7.52.3 Member Function Documentation

#### 7.52.3.1 Dump()

```
bool GazeUtilityLibrary.ScreenArea.Dump (
    string path,
    string prefix ) [inline]
```

Dump the four screen corner points to a csv file

## Parameters

<i>path</i>	The folder to store the file.
<i>prefix</i>	The file prefix.

## Returns

#### 7.52.3.2 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

<i>gazeOrigin</i>	The origin of the gaze.
<i>gazeDirection</i>	The direction of the gaze.

**Returns**

The intersection point with the screen or null if no intersection point exists.

**7.52.3.3 GetPoint2d()**

```
Vector2 GazeUtilityLibrary.ScreenArea.GetPoint2d (
    Vector3 point ) [inline]
```

Get the 2d point on the screen given given a 3d point on the screen plane.

**Parameters**

<i>point</i>	The 3d point on the screen plane to convert.
--------------	--

**Returns**

The 2d point on the screen plane

**7.52.3.4 GetPoint2dNormalized()**

```
Vector2 GazeUtilityLibrary.ScreenArea.GetPoint2dNormalized (
    Vector3 point3d ) [inline]
```

Get the normalized 2d point on the screen given given a 3d point on the screen plane. Note that values outside of the interval [0, 1] indicate an intersection point outside of the screen area.

**Parameters**

<i>point</i>	The 3d point on the screen plane to convert.
--------------	--

**Returns**

The normalized 2d point on the screen plane

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

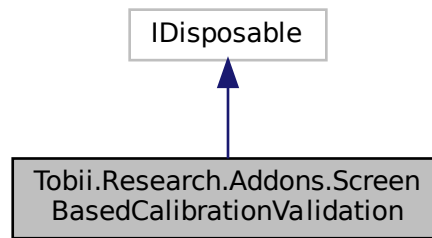
- source/GazeUtilityLibrary/ScreenArea.cs

## 7.53 Tobii.Research.Addons.ScreenBasedCalibrationValidation Class Reference

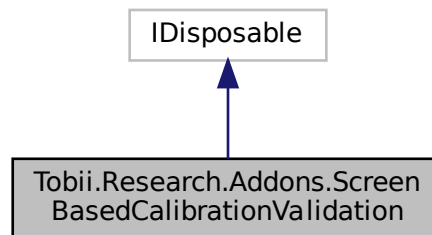
Provides methods and properties for managing calibration validation for screen based eye trackers.



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](#) ()

## 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.

## 7.53.1 Detailed Description

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

## 7.53.2 Member Enumeration Documentation

### 7.53.2.1 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.

## 7.53.3 Constructor & Destructor Documentation

### 7.53.3.1 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

<i>eyeTracker</i>	An IEyeTracker instance.
<i>sampleCount</i>	The number of samples to collect. Default 30, minimum 10, maximum 3000.
<i>timeoutMS</i>	Timeout in milliseconds. Default 1000, minimum 100, maximum 3000.

## 7.53.4 Member Function Documentation

### 7.53.4.1 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](#)

### 7.53.4.2 DiscardData()

```
void Tobii.Research.Addons.ScreenBasedCalibrationValidation.DiscardData (
    NormalizedPoint2D calibrationPointCoordinates ) [inline]
```

Removes the collected data for a specific calibration validation point.

## Parameters

<i>calibrationPointCoordinates</i>	The calibration point to remove.
------------------------------------	----------------------------------

### 7.53.4.3 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.NotInValidationMode

#### 7.53.4.4 EnterValidationMode()

```
void Tobii.Research.Addons.ScreenBasedCalibrationValidation.EnterValidationMode ( ) [inline]
```

Enter the calibration validation mode and starts subscribing to gaze data from the eye tracker.

#### 7.53.4.5 LeaveValidationMode()

```
void Tobii.Research.Addons.ScreenBasedCalibrationValidation.LeaveValidationMode ( ) [inline]
```

Leaves the calibration validation mode, clears all collected data, and unsubscribes from the eye tracker.

#### 7.53.4.6 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

<i>calibrationPointCoordinates</i>	The normalized 2D point on the display area
------------------------------------	---

### 7.53.5 Property Documentation

#### 7.53.5.1 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.

#### 7.53.5.2 State

```
ValidationState Tobii.Research.Addons.ScreenBasedCalibrationValidation.State [get]
```

Get the current state of the validation object.

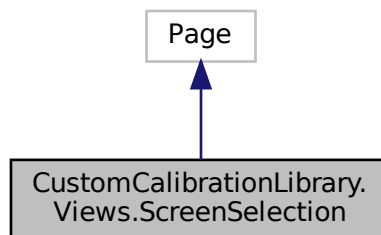
The documentation for this class was generated from the following file:

- `source/TobiiProSdkAddons/ScreenBasedCalibrationValidation.cs`

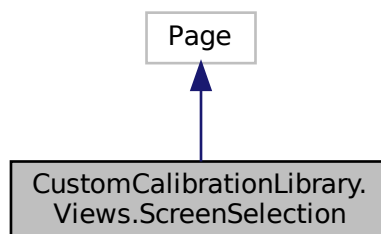
## 7.54 CustomCalibrationLibrary.Views.ScreenSelection Class Reference

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)

#### 7.54.1 Detailed Description

Interaction logic for ScreenSelection.xaml

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

- `source/CustomCalibrationLibrary/Views/ScreenSelection.xaml.cs`

## 7.55 CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel Class Reference

### Public Member Functions

- **ScreenSelectionViewModel** ([CalibrationModel](#) model, Window window)
- void **SwitchScreen** (int index)

### Properties

- ObservableCollection< [Monitor](#) > **Monitors** [get]
- ICommand [CalibrationStartCommand](#) [get]  
*Command to start the calibration*
- ICommand [CalibrationAbortCommand](#) [get]  
*Command to abort the calibration*

### 7.55.1 Property Documentation

#### 7.55.1.1 CalibrationAbortCommand

ICommand CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.CalibrationAbortCommand  
[get]

Command to abort the calibration

#### 7.55.1.2 CalibrationStartCommand

ICommand CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.CalibrationStartCommand  
[get]

Command to start the calibration

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

- source/CustomCalibrationLibrary/ViewModels/ScreenSelectionViewModel.cs

## 7.56 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)
- Vector3? [GetIntersectionPoint](#) (Vector3 origin, Vector3 direction)  
*Compute the intersection point with the triangle with the Moller-Trumbore algorithm.*

## Properties

- Vector3 **V1** [get]
- Vector3 **V2** [get]
- Vector3 **V3** [get]
- Vector3 **E1** [get]
- Vector3 **E2** [get]

### 7.56.1 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.

### 7.56.2 Member Function Documentation

#### 7.56.2.1 GetIntersectionPoint()

```
Vector3? GazeUtilityLibrary.ScreenTriangle.GetIntersectionPoint (
    Vector3 origin,
    Vector3 direction ) [inline]
```

Compute the intersection point with the triangle with the Moller-Trumbore algorithm.

#### Parameters

<i>origin</i>	The origin of the gaze point
<i>direction</i>	The direction of the gaze point

#### Returns

The intersection point or null if no intersection point could be computed.

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

- source/GazeUtilityLibrary/ScreenTriangle.cs

## 7.57 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*

### 7.57.1 Detailed Description

Simple logger class.

### 7.57.2 Constructor & Destructor Documentation

#### 7.57.2.1 TrackerLogger()

```
GazeUtilityLibrary.TrackerLogger.TrackerLogger (
    string? logPath,
    EOutputType type = EOutputType.gaze ) [inline]
```

Initializes a new instance of the [TrackerLogger](#) class.

### 7.57.3 Member Function Documentation

#### 7.57.3.1 Debug()

```
void GazeUtilityLibrary.TrackerLogger.Debug (
    string message ) [inline]
```

wrapper function for debug level logging.

#### Parameters

<i>message</i>	The message.
----------------	--------------



### 7.57.3.2 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

<i>e</i>	The exception.
----------	----------------

### 7.57.3.3 Error()

```
void GazeUtilityLibrary.TrackerLogger.Error (
    string message ) [inline]
```

wrapper function for error level logging

#### Parameters

<i>message</i>	The message.
----------------	--------------

### 7.57.3.4 Info()

```
void GazeUtilityLibrary.TrackerLogger.Info (
    string message ) [inline]
```

wrapper function for info level logging

#### Parameters

<i>message</i>	The message.
----------------	--------------

### 7.57.3.5 Warning()

```
void GazeUtilityLibrary.TrackerLogger.Warning (
    string message ) [inline]
```

wrapper function for warning level logging

## Parameters

<i>message</i>	The message.
----------------	--------------

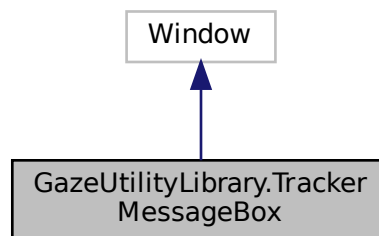
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Logger.cs

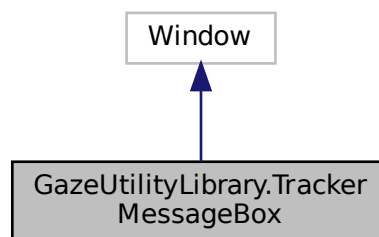
## 7.58 GazeUtilityLibrary.TrackerMessageBox Class Reference

Interaction logic for TrackerMessageBox.xaml

Inheritance diagram for GazeUtilityLibrary.TrackerMessageBox:



Collaboration diagram for GazeUtilityLibrary.TrackerMessageBox:



### 7.58.1 Detailed Description

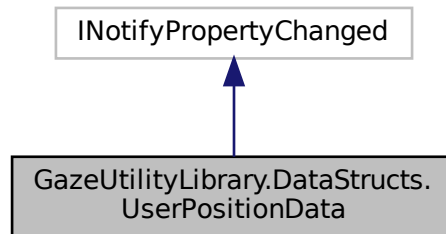
Interaction logic for TrackerMessageBox.xaml

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

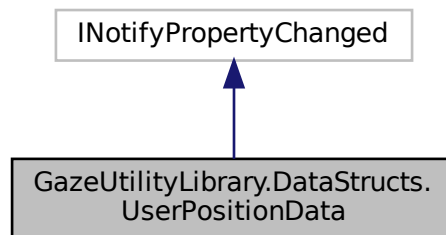
- source/GazeUtilityLibrary/TrackerMessageBox.xaml.cs

## 7.59 GazeUtilityLibrary.DataStructs.UserPositionData Class Reference

Inheritance diagram for GazeUtilityLibrary.DataStructs.UserPositionData:



Collaboration diagram for GazeUtilityLibrary.DataStructs.UserPositionData:



### Public Member Functions

- **UserPositionData** (double xCoordLeft, double yCoordLeft, double zCoordLeft, double xCoordRight, double yCoordRight, double zCoordRight)

### Properties

- double **XCoordLeft** [get, set]
- double **YCoordLeft** [get, set]
- double **ZCoordLeft** [get, set]
- double **XCoordRight** [get, set]
- double **YCoordRight** [get, set]
- double **ZCoordRight** [get, set]

## Events

- PropertyChangedEventHandler? **PropertyChanged**

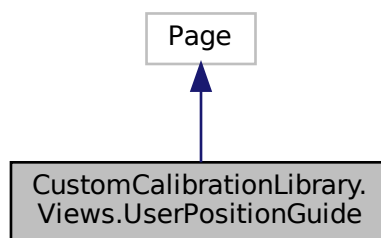
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/UserPositionData.cs

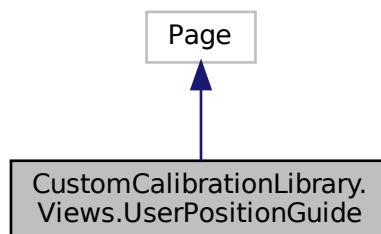
## 7.60 CustomCalibrationLibrary.Views.UserPositionGuide Class Reference

Interaction logic for UserPositionGuide.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.UserPositionGuide:



Collaboration diagram for CustomCalibrationLibrary.Views.UserPositionGuide:



## Public Member Functions

- **UserPositionGuide** ([CalibrationModel](#) model)

### 7.60.1 Detailed Description

Interaction logic for UserPositionGuide.xaml

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

- `source/CustomCalibrationLibrary/Views/UserPositionGuide.xaml.cs`

## 7.61 CustomCalibrationLibrary.ViewModels.UserPositionGuideView↵ Model Class Reference

The view model class for the user position guide view.

### 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*

### 7.61.1 Detailed Description

The view model class for the user position guide view.

### 7.61.2 Constructor & Destructor Documentation

#### 7.61.2.1 UserPositionGuideViewModel()

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

Constructor

## Parameters

<i>model</i>	The calibartion model
--------------	-----------------------

### 7.61.3 Property Documentation

#### 7.61.3.1 CalibrationAbortCommand

`ICommand CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel.CalibrationAbort↔  
Command [get]`

Command to abort the calibration

#### 7.61.3.2 CalibrationStartCommand

`ICommand CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel.CalibrationStart↔  
Command [get]`

Command to start the calibration

#### 7.61.3.3 UserPosition

`UserPositionData CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel.UserPosition  
[get]`

The user position to be represented on the view

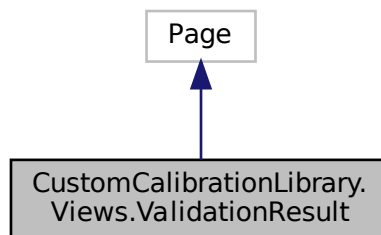
The documentation for this class was generated from the following file:

- `source/CustomCalibrationLibrary/ViewModels/UserPositionGuideViewModel.cs`

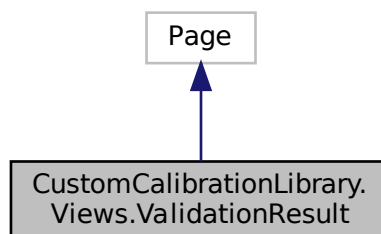
## 7.62 CustomCalibrationLibrary.Views.ValidationResult Class Reference

Interaction logic for ValidationResult.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.ValidationResult:



Collaboration diagram for CustomCalibrationLibrary.Views.ValidationResult:



### Public Member Functions

- **ValidationResult** ([CalibrationModel](#) model)

#### 7.62.1 Detailed Description

Interaction logic for ValidationResult.xaml

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

- `source/CustomCalibrationLibrary/Views/ValidationResult.xaml.cs`

## 7.63 CustomCalibrationLibrary.ViewModels.ValidationResultViewModel Class Reference

View model class of the gaze validation result.

### 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*

#### 7.63.1 Detailed Description

View model class of the gaze validation result.

#### 7.63.2 Constructor & Destructor Documentation

##### 7.63.2.1 ValidationResultViewModel()

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

Constructor

Parameters

<i>model</i>	The claibration model
--------------	-----------------------

#### 7.63.3 Property Documentation



### 7.63.3.1 ValidationCloseCommand

```
 ICommand CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.ValidationCloseCommand  
 [get]
```

Command to close the validation window

### 7.63.3.2 ValidationData

```
 GazeValidationData CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.Validation↔  
 Data [get]
```

The validation result

### 7.63.3.3 ValidationRestartCommand

```
 ICommand CustomCalibrationLibrary.ViewModels.ValidationResultViewModel.ValidationRestart↔  
 Command [get]
```

Command to restart the validation

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

- source/CustomCalibrationLibrary/ViewModels/ValidationResultViewModel.cs



# Index

- AccuracyLeftEye
  - Tobii.Research.Addons.CalibrationValidationPoint, [71](#)
- AccuracyRightEye
  - Tobii.Research.Addons.CalibrationValidationPoint, [71](#)
- App
  - GazeToMouse.App, [31](#)
- ApplyCalibration
  - GazeUtilityLibrary.Tracker.BaseTracker, [38](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [90](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [121](#)
- AverageAccuracyLeftEye
  - Tobii.Research.Addons.CalibrationValidationResult, [73](#)
- AverageAccuracyRightEye
  - Tobii.Research.Addons.CalibrationValidationResult, [74](#)
- AveragePrecisionLeftEye
  - Tobii.Research.Addons.CalibrationValidationResult, [74](#)
- AveragePrecisionRightEye
  - Tobii.Research.Addons.CalibrationValidationResult, [74](#)
- AveragePrecisionRMSLeftEye
  - Tobii.Research.Addons.CalibrationValidationResult, [74](#)
- AveragePrecisionRMSRightEye
  - Tobii.Research.Addons.CalibrationValidationResult, [74](#)
- BaseTracker
  - GazeUtilityLibrary.Tracker.BaseTracker, [36](#)
- CalibrationAbortCommand
  - CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel, [136](#)
  - CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel, [144](#)
  - CustomCalibrationLibrary.Views.CalibrationFailed, [54](#)
  - CustomCalibrationLibrary.Views.Disconnect, [81](#)
- CalibrationAcceptCommand
  - CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, [70](#)
- CalibrationEvent
  - CustomCalibrationLibrary.Models.CalibrationModel, [60](#)
- CalibrationEventType
  - CustomCalibrationLibrary.Models, [20](#)
- CalibrationFailed
  - CustomCalibrationLibrary.Views.CalibrationFailed, [53](#)
- CalibrationOutputValue
  - GazeUtilityLibrary.DataStructs, [24](#)
- CalibrationPoints
  - CustomCalibrationLibrary.Models.CalibrationModel, [59](#)
  - CustomCalibrationLibrary.ViewModels.CalibrationViewModel, [76](#)
- CalibrationRestartCommand
  - CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, [70](#)
  - CustomCalibrationLibrary.Views.CalibrationFailed, [54](#)
- CalibrationResultViewModel
  - CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, [69](#)
- CalibrationStartCommand
  - CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel, [136](#)
  - CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel, [144](#)
- CalibrationStatus
  - CustomCalibrationLibrary.Models, [20](#)
- CalibrationValidate
  - GazeToMouse.App, [31](#)
- CalibrationViewModel
  - CustomCalibrationLibrary.ViewModels.CalibrationViewModel, [76](#)
- CleanupCalibrationOutputFile
  - GazeUtilityLibrary.GazeConfiguration, [99](#)
- CleanupGazeOutputFile
  - GazeUtilityLibrary.GazeConfiguration, [100](#)
- CleanupValidationOutputFile
  - GazeUtilityLibrary.GazeConfiguration, [100](#)
- CollectCalibrationDataAsync
  - GazeUtilityLibrary.Tracker.BaseTracker, [38](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [90](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [121](#)
- CollectValidationDataAsync
  - GazeUtilityLibrary.Tracker.BaseTracker, [38](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [91](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [121](#)
- CompensateDrift
  - GazeToMouse.App, [31](#)
- Compensation
  - GazeUtilityLibrary.DataStructs.DriftCompensationData, [84](#)

- Compute
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 133
- ComputeValidation
  - GazeUtilityLibrary.Tracker.BaseTracker, 40
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 91
  - GazeUtilityLibrary.Tracker.MouseTracker, 122
- config
  - GazeUtilityLibrary.Tracker.BaseTracker, 46
- ConvertToBinString
  - GazeUtilityLibrary.GazeError, 111
- Coordinates
  - Tobii.Research.Addons.CalibrationValidationPoint, 71
- CustomCalibrate
  - GazeToMouse.App, 32
- CustomCalibrationLibrary, 19
- CustomCalibrationLibrary.Commands, 19
- CustomCalibrationLibrary.Commands.CalibrationCommand, 50
- CustomCalibrationLibrary.Converters, 19
- CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter, 112
- CustomCalibrationLibrary.Converters.PositionConverter, 125
  - OffsetProperty, 126
- CustomCalibrationLibrary.Converters.ProximityColorConverter, 127
- CustomCalibrationLibrary.Models, 19
  - CalibrationEventType, 20
  - CalibrationStatus, 20
- CustomCalibrationLibrary.Models.CalibrationModel, 56
  - CalibrationEvent, 60
  - CalibrationPoints, 59
  - Error, 59
  - GazeDataCollected, 57
  - GazePoint, 59
  - Index, 59
  - InitCalibration, 57
  - LastStatus, 59
  - NextCalibrationPoint, 58
  - Points, 59
  - RedoCalibrationPoint, 58
  - SetCalibrationResult, 58
  - Status, 60
  - UpdateGazePoint, 58
  - UserPositionGuide, 60
- CustomCalibrationLibrary.ViewModels, 20
- CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel, 64
- CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, 68
  - CalibrationAcceptCommand, 70
  - CalibrationRestartCommand, 70
  - CalibrationResultViewModel, 69
  - GazePoint, 70
  - GazeVisibilityCommand, 70
  - OnGazeToggle, 69
- CustomCalibrationLibrary.ViewModels.CalibrationViewModel, 75
  - CalibrationPoints, 76
  - CalibrationViewModel, 76
- CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel, 84
  - DriftCompensationViewModel, 85
  - FixationPoint, 85
- CustomCalibrationLibrary.ViewModels.Monitor, 117
- CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel, 136
  - CalibrationAbortCommand, 136
  - CalibrationStartCommand, 136
- CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel, 143
  - CalibrationAbortCommand, 144
  - CalibrationStartCommand, 144
  - UserPosition, 144
  - UserPositionGuideViewModel, 143
- CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, 146
  - ValidationCloseCommand, 146
  - ValidationData, 147
  - ValidationRestartCommand, 147
  - ValidationResultViewModel, 146
- CustomCalibrationLibrary.Views, 21
  - CustomCalibrationLibrary.Views.Calibration, 49
  - CustomCalibrationLibrary.Views.CalibrationFailed, 52
    - CalibrationAbortCommand, 54
    - CalibrationFailed, 53
    - CalibrationRestartCommand, 54
    - Error, 54
    - PropertyChanged, 54
  - CustomCalibrationLibrary.Views.CalibrationFrame, 55
  - CustomCalibrationLibrary.Views.CalibrationPoint, 63
  - CustomCalibrationLibrary.Views.CalibrationResult, 65
  - CustomCalibrationLibrary.Views.CalibrationResultPoint, 67
  - CustomCalibrationLibrary.Views.CalibrationWindow, 77
  - CustomCalibrationLibrary.Views.Computing, 78
  - CustomCalibrationLibrary.Views.Disconnect, 80
    - CalibrationAbortCommand, 81
  - CustomCalibrationLibrary.Views.DriftCompensationWindow, 86
  - CustomCalibrationLibrary.Views.FixationPoint, 95
  - CustomCalibrationLibrary.Views.ScreenSelection, 135
  - CustomCalibrationLibrary.Views.UserPositionGuide, 142
  - CustomCalibrationLibrary.Views.ValidationResult, 145
- CustomCalibrationLibrary.Views.ValidationResultModel, 145
  - Debug
- GazeUtilityLibrary.TrackerLogger, 138
  - DeviceName
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
  - dialogBoxTimer
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
  - DiscardData
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 133

- Dispose
  - GazeUtilityLibrary.Tracker.BaseTracker, 40
  - GazeUtilityLibrary.Tracker.MouseTracker, 122
  - Tobii.Research.Addons.ScreenBasedCalibrationValidationControl, 21
- driftCompensation
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
- DriftCompensationComputed
  - GazeUtilityLibrary.Tracker.BaseTracker, 48
- DriftCompensationData
  - GazeUtilityLibrary.DataStructs.DriftCompensationData, 83
- DriftCompensationEventHandler
  - GazeUtilityLibrary.Tracker.BaseTracker, 41
- DriftCompensationViewModel
  - CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel, 85
- Dump
  - GazeUtilityLibrary.ScreenArea, 129
- DumpCurrentConfigurationFile
  - GazeUtilityLibrary.GazeConfiguration, 100
- DumpFatal
  - GazeUtilityLibrary.TrackerLogger, 139
- ECalibrationDataError
  - GazeUtilityLibrary, 23
- EGazeConfigError
  - GazeUtilityLibrary, 23
- EGazeDataError
  - GazeUtilityLibrary, 23
- EnterValidationMode
  - Tobii.Research.Addons.ScreenBasedCalibrationValidationControl, 133
- Error
  - CustomCalibrationLibrary.Models.CalibrationModel, 59
  - CustomCalibrationLibrary.Views.CalibrationFailed, 54
  - GazeUtilityLibrary.TrackerLogger, 139
- EyeData
  - GazeUtilityLibrary.DataStructs.EyeData, 87
- EyeTrackerPro
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 90
- FinishCalibration
  - GazeUtilityLibrary.Tracker.BaseTracker, 41
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 91
  - GazeUtilityLibrary.Tracker.MouseTracker, 122
- FinishCalibrationAsync
  - GazeUtilityLibrary.Tracker.BaseTracker, 41
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 92
  - GazeUtilityLibrary.Tracker.MouseTracker, 122
- FinishValidation
  - GazeUtilityLibrary.Tracker.BaseTracker, 41
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 92
  - GazeUtilityLibrary.Tracker.MouseTracker, 123
- FixationPoint
  - CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel, 85
- GazeCalibrationData
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, 96
- GazeControl
  - GazeControl.App, 27
- GazeData
  - GazeUtilityLibrary.DataStructs.GazeData, 103, 104
  - Tobii.Research.Addons.CalibrationValidationPoint, 72
- GazeData2d
  - GazeUtilityLibrary.DataStructs.GazeData2d, 106
- GazeData3d
  - GazeUtilityLibrary.DataStructs.GazeData3d, 107
- GazeDataCollected
  - CustomCalibrationLibrary.Models.CalibrationModel, 59
- GazeDataCollection
  - GazeUtilityLibrary.DataStructs.GazeDataCollection, 108
- GazeDataHandler
  - GazeUtilityLibrary.Tracker.BaseTracker, 41
- GazeDataReceived
  - GazeUtilityLibrary.Tracker.BaseTracker, 48
- GazeOutputValue
  - GazeUtilityLibrary.DataStructs, 24
- GazePoint
  - CustomCalibrationLibrary.Models.CalibrationModel, 59
  - CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, 70
- GazePosition2d
  - GazeUtilityLibrary.DataStructs.DriftCompensationData, 84
- GazePosition3d
  - GazeUtilityLibrary.DataStructs.DriftCompensationData, 84
- GazePositionAverage
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, 62
- GazePositionLeft
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, 62
- GazePositionRight
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, 62
- GazeRecordingDisable
  - GazeToMouse.App, 32
- GazeRecordingEnable
  - GazeToMouse.App, 32
- GazeToMouse, 21
- GazeToMouse.App, 30
  - App, 31
  - CalibrationValidate, 31
  - CompensateDrift, 31
  - CustomCalibrate, 32
  - GazeRecordingDisable, 32
  - GazeRecordingEnable, 32
  - MouseTrackingDisable, 32
  - MouseTrackingEnable, 32
- ResetDrift
  - CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel, 85
  - GazeUtilityLibrary, 22

- ECalibrationDataError, [23](#)
  - EGazeConfigError, [23](#)
  - EGazeDataError, [23](#)
- GazeUtilityLibrary.CalibrationDataError, [51](#)
  - GetCalibrationDataErrorString, [52](#)
- GazeUtilityLibrary.ConfigItem, [78](#)
- GazeUtilityLibrary.ConfigScreenArea, [79](#)
- GazeUtilityLibrary.DataStructs, [23](#)
  - CalibrationOutputValue, [24](#)
  - GazeOutputValue, [24](#)
  - ValidationOutputValue, [24](#)
- GazeUtilityLibrary.DataStructs.CalibrationPoint, [61](#)
  - GazePositionAverage, [62](#)
  - GazePositionLeft, [62](#)
  - GazePositionRight, [62](#)
  - HasData, [62](#)
  - Index, [63](#)
  - Position, [63](#)
- GazeUtilityLibrary.DataStructs.DriftCompensationData, [83](#)
  - Compensation, [84](#)
  - DriftCompensationData, [83](#)
  - GazePosition2d, [84](#)
  - GazePosition3d, [84](#)
- GazeUtilityLibrary.DataStructs.EyeData, [86](#)
  - EyeData, [87](#)
  - IsPupilDiameterValid, [87](#)
  - PupilDiameter, [87](#)
- GazeUtilityLibrary.DataStructs.GazeCalibrationData, [95](#)
  - GazeCalibrationData, [96](#)
  - Prepare, [97](#)
- GazeUtilityLibrary.DataStructs.GazeData, [103](#)
  - GazeData, [103](#), [104](#)
  - Prepare, [105](#)
- GazeUtilityLibrary.DataStructs.GazeData2d, [106](#)
  - GazeData2d, [106](#)
- GazeUtilityLibrary.DataStructs.GazeData3d, [106](#)
  - GazeData3d, [107](#)
- GazeUtilityLibrary.DataStructs.GazeDataCollection, [107](#)
  - GazeDataCollection, [108](#)
- GazeUtilityLibrary.DataStructs.GazeValidationData, [112](#)
  - Prepare, [112](#)
- GazeUtilityLibrary.DataStructs.LiveGazePoint, [116](#)
- GazeUtilityLibrary.DataStructs.PipeCommand, [125](#)
- GazeUtilityLibrary.DataStructs.UserPositionData, [141](#)
- GazeUtilityLibrary.DriftCompensation, [81](#)
  - Q, [83](#)
  - Reset, [82](#)
  - Start, [82](#)
  - Update, [82](#)
- GazeUtilityLibrary.GazeConfigError, [97](#)
  - GetGazeConfigErrorString, [98](#)
- GazeUtilityLibrary.GazeConfiguration, [99](#)
  - CleanupCalibrationOutputFile, [99](#)
  - CleanupGazeOutputFile, [100](#)
  - CleanupValidationOutputFile, [100](#)
  - DumpCurrentConfigurationFile, [100](#)
  - InitConfig, [100](#)
  - PrepareCalibrationOutputFile, [101](#)
  - PrepareGazeOutputFile, [101](#)
  - PrepareValidationOutputFile, [101](#)
  - WriteToCalibrationOutput, [102](#)
  - WriteToGazeOutput, [102](#)
  - WriteToValidationOutput, [102](#)
- GazeUtilityLibrary.GazeDataError, [109](#)
  - GetGazeDataErrorString, [110](#)
- GazeUtilityLibrary.GazeError, [111](#)
  - ConvertToBinString, [111](#)
- GazeUtilityLibrary.JsonConfigParser, [114](#)
  - GetDefaultConfig, [114](#)
  - JsonConfigParser, [114](#)
  - ParseJsonConfig, [115](#)
  - SerializeJsonConfig, [115](#)
- GazeUtilityLibrary.MouseHider, [117](#)
  - HideCursor, [118](#)
  - MouseHider, [117](#)
  - ShowCursor, [118](#)
- GazeUtilityLibrary.ScreenArea, [128](#)
  - Dump, [129](#)
  - GetIntersectionPoint, [129](#)
  - GetPoint2d, [130](#)
  - GetPoint2dNormalized, [130](#)
  - ScreenArea, [128](#)
- GazeUtilityLibrary.ScreenTriangle, [136](#)
  - GetIntersectionPoint, [137](#)
- GazeUtilityLibrary.Tracker, [25](#)
- GazeUtilityLibrary.Tracker.BaseTracker, [33](#)
  - ApplyCalibration, [38](#)
  - BaseTracker, [36](#)
  - CollectCalibrationDataAsync, [38](#)
  - CollectValidationDataAsync, [38](#)
  - ComputeValidation, [40](#)
  - config, [46](#)
  - DeviceName, [47](#)
  - dialogBoxTimer, [47](#)
  - Dispose, [40](#)
  - driftCompensation, [47](#)
  - DriftCompensationComputed, [48](#)
  - DriftCompensationEventHandler, [41](#)
  - FinishCalibration, [41](#)
  - FinishCalibrationAsync, [41](#)
  - FinishValidation, [41](#)
  - GazeDataHandler, [41](#)
  - GazeDataReceived, [48](#)
  - GetFixationFrameCount, [42](#)
  - GetUnitDirection, [42](#)
  - InitCalibration, [42](#)
  - InitCalibrationAsync, [42](#)
  - InitDriftCompensation, [43](#)
  - InitValidation, [43](#)
  - IsInitialised, [43](#)
  - IsReady, [43](#)
  - logger, [47](#)
  - OnGazeDataReceived, [44](#)
  - OnPropertyChanged, [44](#)
  - OnTrackerDisabled, [44](#)

- OnTrackerDisabledTimeout, [45](#)
- OnTrackerEnabled, [45](#)
- OnUserPositionDataReceived, [45](#)
- PatternReplace, [45](#)
- PropertyChanged, [48](#)
- ResetDriftCompensation, [46](#)
- screenArea, [47](#)
- StartDriftCompensation, [46](#)
- State, [48](#)
- TrackerDisabled, [48](#)
- TrackerEnabled, [48](#)
- trackerMessageBox, [47](#)
- UserPositionDataHandler, [46](#)
- UserPositionDataReceived, [49](#)
- GazeUtilityLibrary.Tracker.EyeTrackerPro, [88](#)
  - ApplyCalibration, [90](#)
  - CollectCalibrationDataAsync, [90](#)
  - CollectValidationDataAsync, [91](#)
  - ComputeValidation, [91](#)
  - EyeTrackerPro, [90](#)
  - FinishCalibration, [91](#)
  - FinishCalibrationAsync, [92](#)
  - FinishValidation, [92](#)
  - GetFixationFrameCount, [92](#)
  - GetUnitDirection, [92](#)
  - InitCalibration, [93](#)
  - InitCalibrationAsync, [93](#)
  - InitDriftCompensation, [93](#)
  - InitValidation, [93](#)
  - IsInitialised, [94](#)
  - IsLicenseOk, [94](#)
  - PatternReplace, [94](#)
- GazeUtilityLibrary.Tracker.MouseTracker, [118](#)
  - ApplyCalibration, [121](#)
  - CollectCalibrationDataAsync, [121](#)
  - CollectValidationDataAsync, [121](#)
  - ComputeValidation, [122](#)
  - Dispose, [122](#)
  - FinishCalibration, [122](#)
  - FinishCalibrationAsync, [122](#)
  - FinishValidation, [123](#)
  - GetFixationFrameCount, [123](#)
  - GetUnitDirection, [123](#)
  - InitCalibration, [123](#)
  - InitCalibrationAsync, [124](#)
  - InitDriftCompensation, [124](#)
  - InitValidation, [124](#)
  - MouseTracker, [120](#)
  - Start, [124](#)
  - Stop, [124](#)
- GazeUtilityLibrary.TrackerLogger, [137](#)
  - Debug, [138](#)
  - DumpFatal, [139](#)
  - Error, [139](#)
  - Info, [139](#)
  - TrackerLogger, [138](#)
  - Warning, [139](#)
- GazeUtilityLibrary.TrackerMessageBox, [140](#)
- GazeVisibilityCommand
  - CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, [70](#)
- GetCalibrationDataErrorString
  - GazeUtilityLibrary.CalibrationDataError, [52](#)
- GetDefaultConfig
  - GazeUtilityLibrary.JsonConfigParser, [114](#)
- GetFixationFrameCount
  - GazeUtilityLibrary.Tracker.BaseTracker, [42](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [92](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [123](#)
- GetGazeConfigErrorString
  - GazeUtilityLibrary.GazeConfigError, [98](#)
- GetGazeDataErrorString
  - GazeUtilityLibrary.GazeDataError, [110](#)
- GetIntersectionPoint
  - GazeUtilityLibrary.ScreenArea, [129](#)
  - GazeUtilityLibrary.ScreenTriangle, [137](#)
- GetPoint2d
  - GazeUtilityLibrary.ScreenArea, [130](#)
- GetPoint2dNormalized
  - GazeUtilityLibrary.ScreenArea, [130](#)
- GetUnitDirection
  - GazeUtilityLibrary.Tracker.BaseTracker, [42](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [92](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [123](#)
- HasData
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, [62](#)
- HideCursor
  - GazeUtilityLibrary.MouseHider, [118](#)
- Index
  - CustomCalibrationLibrary.Models.CalibrationModel, [59](#)
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, [63](#)
- Info
  - GazeUtilityLibrary.TrackerLogger, [139](#)
- InitCalibration
  - CustomCalibrationLibrary.Models.CalibrationModel, [57](#)
  - GazeUtilityLibrary.Tracker.BaseTracker, [42](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [93](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [123](#)
- InitCalibrationAsync
  - GazeUtilityLibrary.Tracker.BaseTracker, [42](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [93](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [124](#)
- InitConfig
  - GazeUtilityLibrary.GazeConfiguration, [100](#)
- InitDriftCompensation
  - GazeUtilityLibrary.Tracker.BaseTracker, [43](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [93](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [124](#)
- InitValidation
  - GazeUtilityLibrary.Tracker.BaseTracker, [43](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [93](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [124](#)
- IsInitialised



- GazeUtilityLibrary.Tracker.BaseTracker, 43
- GazeUtilityLibrary.Tracker.EyeTrackerPro, 94
- IsLicenseOk
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 94
- IsPupilDiameterValid
  - GazeUtilityLibrary.DataStructs.EyeData, 87
- IsReady
  - GazeUtilityLibrary.Tracker.BaseTracker, 43
- JsonConfigParser
  - GazeUtilityLibrary.JsonConfigParser, 114
- LastStatus
  - CustomCalibrationLibrary.Models.CalibrationModel, 59
- LeaveValidationMode
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 134
- logger
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
- MouseHider
  - GazeUtilityLibrary.MouseHider, 117
- MouseTracker
  - GazeUtilityLibrary.Tracker.MouseTracker, 120
- MouseTrackingDisable
  - GazeToMouse.App, 32
- MouseTrackingEnable
  - GazeToMouse.App, 32
- NextCalibrationPoint
  - CustomCalibrationLibrary.Models.CalibrationModel, 58
- OffsetProperty
  - CustomCalibrationLibrary.Converters.PositionConverter, 126
- OnGazeDataReceived
  - GazeUtilityLibrary.Tracker.BaseTracker, 44
- OnGazeToggle
  - CustomCalibrationLibrary.ViewModels.CalibrationResponseModel, 69
- OnPropertyChanged
  - GazeUtilityLibrary.Tracker.BaseTracker, 44
- OnTrackerDisabled
  - GazeUtilityLibrary.Tracker.BaseTracker, 44
- OnTrackerDisabledTimeout
  - GazeUtilityLibrary.Tracker.BaseTracker, 45
- OnTrackerEnabled
  - GazeUtilityLibrary.Tracker.BaseTracker, 45
- OnUserPositionDataReceived
  - GazeUtilityLibrary.Tracker.BaseTracker, 45
- ParseJsonConfig
  - GazeUtilityLibrary.JsonConfigParser, 115
- PatternReplace
  - GazeUtilityLibrary.Tracker.BaseTracker, 45
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 94
- Points
  - CustomCalibrationLibrary.Models.CalibrationModel, 59
  - Tobii.Research.Addons.CalibrationValidationResult, 74
- Position
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, 63
- PrecisionLeftEye
  - Tobii.Research.Addons.CalibrationValidationPoint, 72
- PrecisionRightEye
  - Tobii.Research.Addons.CalibrationValidationPoint, 72
- PrecisionRMSLeftEye
  - Tobii.Research.Addons.CalibrationValidationPoint, 72
- PrecisionRMSRightEye
  - Tobii.Research.Addons.CalibrationValidationPoint, 72
- Prepare
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, 97
  - GazeUtilityLibrary.DataStructs.GazeData, 105
  - GazeUtilityLibrary.DataStructs.GazeValidationData, 112
- PrepareCalibrationOutputFile
  - GazeUtilityLibrary.GazeConfiguration, 101
- PrepareGazeOutputFile
  - GazeUtilityLibrary.GazeConfiguration, 101
- PrepareValidationOutputFile
  - GazeUtilityLibrary.GazeConfiguration, 101
- PropertyChanged
  - CustomCalibrationLibrary.Views.CalibrationFailed, 54
  - GazeUtilityLibrary.Tracker.BaseTracker, 48
- PupilDiameter
  - GazeUtilityLibrary.DataStructs.EyeData, 87
- Q
  - GazeUtilityLibrary.DriftCompensation, 83
- RecalibrationPoint
  - CustomCalibrationLibrary.Models.CalibrationModel, 58
- Reset
  - GazeUtilityLibrary.DriftCompensation, 82
- ResetDriftCompensation
  - GazeToMouse.App, 33
  - GazeUtilityLibrary.Tracker.BaseTracker, 46
- Result
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 134
- ScreenArea
  - GazeUtilityLibrary.ScreenArea, 128
- screenArea
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
- ScreenBasedCalibrationValidation
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 132



- SerializeJsonConfig
  - GazeUtilityLibrary.JsonConfigParser, 115
- SetCalibrationResult
  - CustomCalibrationLibrary.Models.CalibrationModel, 58
- ShowCursor
  - GazeUtilityLibrary.MouseHider, 118
- ShowMouse, 25
- ShowMouse.App, 28
- Start
  - GazeUtilityLibrary.DriftCompensation, 82
  - GazeUtilityLibrary.Tracker.MouseTracker, 124
- StartCollectingData
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 134
- StartDriftCompensation
  - GazeUtilityLibrary.Tracker.BaseTracker, 46
- State
  - GazeUtilityLibrary.Tracker.BaseTracker, 48
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 134
- Status
  - CustomCalibrationLibrary.Models.CalibrationModel, 60
- Stop
  - GazeUtilityLibrary.Tracker.MouseTracker, 124
- TimedOut
  - Tobii.Research.Addons.CalibrationValidationPoint, 72
- Tobii, 25
- Tobii.Research, 25
- Tobii.Research.Addons, 25
- Tobii.Research.Addons.CalibrationValidationPoint, 70
  - AccuracyLeftEye, 71
  - AccuracyRightEye, 71
  - Coordinates, 71
  - GazeData, 72
  - PrecisionLeftEye, 72
  - PrecisionRightEye, 72
  - PrecisionRMSLeftEye, 72
  - PrecisionRMSRightEye, 72
  - TimedOut, 72
- Tobii.Research.Addons.CalibrationValidationResult, 73
  - AverageAccuracyLeftEye, 73
  - AverageAccuracyRightEye, 74
  - AveragePrecisionLeftEye, 74
  - AveragePrecisionRightEye, 74
  - AveragePrecisionRMSLeftEye, 74
  - AveragePrecisionRMSRightEye, 74
  - Points, 74
- Tobii.Research.Addons.ScreenBasedCalibrationValidation, 130
  - Compute, 133
  - DiscardData, 133
  - Dispose, 133
  - EnterValidationMode, 133
  - LeaveValidationMode, 134
  - Result, 134
  - ScreenBasedCalibrationValidation, 132
  - StartCollectingData, 134
  - State, 134
  - ValidationState, 132
- Tobii.Research.Addons.Utility, 26
- TobiiCalibrate, 26
- TobiiCalibrate.App, 29
- TrackerDisabled
  - GazeUtilityLibrary.Tracker.BaseTracker, 48
- TrackerEnabled
  - GazeUtilityLibrary.Tracker.BaseTracker, 48
- TrackerLogger
  - GazeUtilityLibrary.TrackerLogger, 138
- TrackerMessageBox
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
- Update
  - GazeUtilityLibrary.DriftCompensation, 82
- UpdateGazePoint
  - CustomCalibrationLibrary.Models.CalibrationModel, 58
- UserPosition
  - CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel, 144
- UserPositionDataHandler
  - GazeUtilityLibrary.Tracker.BaseTracker, 46
- UserPositionDataReceived
  - GazeUtilityLibrary.Tracker.BaseTracker, 49
- UserPositionGuide
  - CustomCalibrationLibrary.Models.CalibrationModel, 60
- UserPositionGuideViewModel
  - CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel, 143
- ValidationCloseCommand
  - CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, 146
- ValidationData
  - CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, 147
- ValidationOutputValue
  - GazeUtilityLibrary.DataStructs, 24
- ValidationRestartCommand
  - CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, 147
- ValidationResultViewModel
  - CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, 146
- ValidationState
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 132
- Warning
  - GazeUtilityLibrary.TrackerLogger, 139
- WriteToCalibrationOutput
  - GazeUtilityLibrary.GazeConfiguration, 102
- WriteToGazeOutput
  - GazeUtilityLibrary.GazeConfiguration, 102

WriteToValidationOutput

GazeUtilityLibrary.GazeConfiguration, [102](#)