

# Gaze Toolset

v3.4.0

Generated by Doxygen 1.8.17



<b>1 Changelog</b>	<b>1</b>
<b>2 Toolset to Control Tobii Eye Tracker</b>	<b>9</b>
<b>3 Sample Files for Experimentation with Eye Tracker Utility</b>	<b>13</b>
<b>4 Namespace Index</b>	<b>15</b>
4.1 Namespace List . . . . .	15
<b>5 Hierarchical Index</b>	<b>17</b>
5.1 Class Hierarchy . . . . .	17
<b>6 Class Index</b>	<b>19</b>
6.1 Class List . . . . .	19
<b>7 Namespace Documentation</b>	<b>23</b>
7.1 CustomCalibrationLibrary Namespace Reference . . . . .	23
7.2 CustomCalibrationLibrary.Commands Namespace Reference . . . . .	23
7.3 CustomCalibrationLibrary.Converters Namespace Reference . . . . .	23
7.4 CustomCalibrationLibrary.Models Namespace Reference . . . . .	23
7.4.1 Enumeration Type Documentation . . . . .	24
7.4.1.1 CalibrationEventType . . . . .	24
7.4.1.2 CalibrationStatus . . . . .	24
7.5 CustomCalibrationLibrary.ViewModels Namespace Reference . . . . .	24
7.6 CustomCalibrationLibrary.Views Namespace Reference . . . . .	25
7.7 GazeControl Namespace Reference . . . . .	25
7.8 GazeToMouse Namespace Reference . . . . .	25
7.9 GazeUtilityLibrary Namespace Reference . . . . .	26
7.9.1 Detailed Description . . . . .	27
7.9.2 Enumeration Type Documentation . . . . .	27
7.9.2.1 ECalibrationDataError . . . . .	27
7.9.2.2 EGazeConfigError . . . . .	27
7.9.2.3 EGazeDataError . . . . .	27
7.9.2.4 EOutputType . . . . .	27
7.10 GazeUtilityLibrary.DataStructs Namespace Reference . . . . .	28
7.10.1 Enumeration Type Documentation . . . . .	29
7.10.1.1 CalibrationOutputValue . . . . .	29
7.10.1.2 GazeOutputValue . . . . .	29
7.10.1.3 ValidationOutputValue . . . . .	29
7.11 GazeUtilityLibrary.Tracker Namespace Reference . . . . .	29
7.12 ShowMouse Namespace Reference . . . . .	30
7.13 Tobii Namespace Reference . . . . .	30
7.14 Tobii.Research Namespace Reference . . . . .	30
7.15 Tobii.Research.Addons Namespace Reference . . . . .	30

7.16 Tobii.Research.Addons.Utility Namespace Reference . . . . .	30
7.17 TobiiCalibrate Namespace Reference . . . . .	30
<b>8 Class Documentation</b>	<b>31</b>
8.1 GazeControl.App Class Reference . . . . .	31
8.1.1 Detailed Description . . . . .	32
8.2 GazeToMouse.App Class Reference . . . . .	32
8.2.1 Detailed Description . . . . .	33
8.2.2 Constructor & Destructor Documentation . . . . .	33
8.2.2.1 App() . . . . .	33
8.2.3 Member Function Documentation . . . . .	34
8.2.3.1 CalibrationValidate() . . . . .	34
8.2.3.2 CompensateDrift() . . . . .	34
8.2.3.3 CustomCalibrate() . . . . .	34
8.2.3.4 GazeRecordingDisable() . . . . .	34
8.2.3.5 GazeRecordingEnable() . . . . .	35
8.2.3.6 MouseTrackingDisable() . . . . .	35
8.2.3.7 MouseTrackingEnable() . . . . .	35
8.2.3.8 ResetDriftCompensation() . . . . .	35
8.2.4 Property Documentation . . . . .	35
8.2.4.1 LastTag . . . . .	35
8.2.4.2 StartTime . . . . .	35
8.2.4.3 Tag . . . . .	36
8.2.4.4 TrialId . . . . .	36
8.3 ShowMouse.App Class Reference . . . . .	36
8.3.1 Detailed Description . . . . .	37
8.4 TobiiCalibrate.App Class Reference . . . . .	37
8.4.1 Detailed Description . . . . .	37
8.5 GazeUtilityLibrary.Tracker.BaseTracker Class Reference . . . . .	38
8.5.1 Detailed Description . . . . .	41
8.5.2 Member Enumeration Documentation . . . . .	41
8.5.2.1 DeviceStatus . . . . .	41
8.5.3 Constructor & Destructor Documentation . . . . .	41
8.5.3.1 BaseTracker() . . . . .	41
8.5.4 Member Function Documentation . . . . .	42
8.5.4.1 ApplyCalibration() . . . . .	42
8.5.4.2 CollectCalibrationDataAsync() . . . . .	42
8.5.4.3 CollectValidationDataAsync() . . . . .	43
8.5.4.4 ComputeValidation() . . . . .	44
8.5.4.5 Dispose() [1/2] . . . . .	44
8.5.4.6 Dispose() [2/2] . . . . .	44
8.5.4.7 DriftCompensationEventHandler() . . . . .	45

8.5.4.8 FinishCalibration()	45
8.5.4.9 FinishCalibrationAsync()	45
8.5.4.10 FinishValidation()	45
8.5.4.11 GazeDataHandler()	45
8.5.4.12 GetFixationFrameCount()	46
8.5.4.13 GetUnitDirection()	46
8.5.4.14 InitCalibration()	46
8.5.4.15 InitCalibrationAsync()	47
8.5.4.16 InitDriftCompensation()	47
8.5.4.17 InitValidation()	47
8.5.4.18 IsInitialised()	47
8.5.4.19 IsReady()	48
8.5.4.20 OnGazeDataReceived()	48
8.5.4.21 OnPropertyChanged()	48
8.5.4.22 OnTrackerDisabled()	48
8.5.4.23 OnTrackerDisabledTimeout()	49
8.5.4.24 OnTrackerEnabled()	49
8.5.4.25 OnUserPositionDataReceived()	49
8.5.4.26 PatternReplace()	49
8.5.4.27 ResetDriftCompensation()	50
8.5.4.28 StartDriftCompensation()	50
8.5.4.29 UserPositionDataHandler()	50
8.5.5 Member Data Documentation	50
8.5.5.1 config	50
8.5.5.2 DeviceName	51
8.5.5.3 dialogBoxTimer	51
8.5.5.4 driftCompensation	51
8.5.5.5 logger	51
8.5.5.6 screenArea	51
8.5.5.7 trackerMessageBox	51
8.5.6 Property Documentation	52
8.5.6.1 ScreenArea	52
8.5.6.2 State	52
8.5.7 Event Documentation	52
8.5.7.1 DriftCompensationComputed	52
8.5.7.2 GazeDataReceived	52
8.5.7.3 PropertyChanged	52
8.5.7.4 TrackerDisabled	53
8.5.7.5 TrackerEnabled	53
8.5.7.6 UserPositionDataReceived	53
8.6 CustomCalibrationLibrary.Views.Calibration Class Reference	53
8.6.1 Detailed Description	54

8.6.2 Constructor & Destructor Documentation	54
8.6.2.1 Calibration()	54
8.7 CustomCalibrationLibrary.Commands.CalibrationCommand Class Reference	55
8.7.1 Detailed Description	56
8.7.2 Constructor & Destructor Documentation	56
8.7.2.1 CalibrationCommand()	56
8.7.3 Member Function Documentation	56
8.7.3.1 CanExecute()	56
8.7.3.2 Execute()	56
8.7.4 Property Documentation	57
8.7.4.1 CanExecuteChanged	57
8.8 GazeUtilityLibrary.CalibrationDataError Class Reference	57
8.8.1 Detailed Description	58
8.8.2 Member Function Documentation	58
8.8.2.1 GetCalibrationDataErrorString()	58
8.8.3 Property Documentation	59
8.8.3.1 Error	59
8.9 CustomCalibrationLibrary.Views.CalibrationFailed Class Reference	59
8.9.1 Detailed Description	60
8.9.2 Constructor & Destructor Documentation	60
8.9.2.1 CalibrationFailed()	60
8.9.3 Property Documentation	60
8.9.3.1 CalibrationAbortCommand	61
8.9.3.2 CalibrationRestartCommand	61
8.9.3.3 Error	61
8.9.4 Event Documentation	61
8.9.4.1 PropertyChanged	61
8.10 CustomCalibrationLibrary.Views.CalibrationFrame Class Reference	62
8.10.1 Detailed Description	62
8.10.2 Constructor & Destructor Documentation	62
8.10.2.1 CalibrationFrame()	63
8.11 CustomCalibrationLibrary.Models.CalibrationModel Class Reference	63
8.11.1 Detailed Description	65
8.11.2 Constructor & Destructor Documentation	65
8.11.2.1 CalibrationModel()	65
8.11.3 Member Function Documentation	65
8.11.3.1 GazeDataCollected()	66
8.11.3.2 InitCalibration()	66
8.11.3.3 NextCalibrationPoint()	66
8.11.3.4 OnCalibrationEvent()	66
8.11.3.5 RedoCalibrationPoint()	66
8.11.3.6 SetCalibrationResult()	66

8.11.3.7 UpdateGazePoint()	67
8.11.4 Property Documentation	67
8.11.4.1 CalibrationPoints	67
8.11.4.2 Error	67
8.11.4.3 GazePoint	67
8.11.4.4 Index	68
8.11.4.5 LastStatus	68
8.11.4.6 Points	68
8.11.4.7 Status	68
8.11.4.8 UserPositionGuide	68
8.11.4.9 ValidationData	68
8.11.5 Event Documentation	69
8.11.5.1 CalibrationEvent	69
8.11.5.2 GazePointChanged	69
8.11.5.3 PropertyChanged	69
8.11.5.4 UserPositionGuideChanged	69
8.12 GazeUtilityLibrary.DataStructs.CalibrationPoint Class Reference	70
8.12.1 Detailed Description	71
8.12.2 Constructor & Destructor Documentation	71
8.12.2.1 CalibrationPoint()	71
8.12.3 Property Documentation	71
8.12.3.1 GazePositionAverage	72
8.12.3.2 GazePositionLeft	72
8.12.3.3 GazePositionRight	72
8.12.3.4 HasData	72
8.12.3.5 Index	72
8.12.3.6 Position	72
8.12.4 Event Documentation	73
8.12.4.1 PropertyChanged	73
8.13 CustomCalibrationLibrary.Views.CalibrationPoint Class Reference	73
8.13.1 Detailed Description	74
8.13.2 Constructor & Destructor Documentation	74
8.13.2.1 CalibrationPoint()	74
8.14 CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel Class Reference	74
8.14.1 Detailed Description	75
8.14.2 Constructor & Destructor Documentation	75
8.14.2.1 CalibrationPointViewModel() [1/2]	75
8.14.2.2 CalibrationPointViewModel() [2/2]	76
8.15 CustomCalibrationLibrary.Views.CalibrationResult Class Reference	76
8.15.1 Detailed Description	77
8.15.2 Constructor & Destructor Documentation	77
8.15.2.1 CalibrationResult()	77

8.16 CustomCalibrationLibrary.Views.CalibrationResultPoint Class Reference	78
8.16.1 Detailed Description	78
8.16.2 Constructor & Destructor Documentation	78
8.16.2.1 CalibrationResultPoint()	79
8.17 CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel Class Reference	79
8.17.1 Detailed Description	80
8.17.2 Constructor & Destructor Documentation	81
8.17.2.1 CalibrationResultViewModel()	81
8.17.3 Member Function Documentation	81
8.17.3.1 OnGazeToggle()	81
8.17.4 Property Documentation	81
8.17.4.1 CalibrationAcceptCommand	81
8.17.4.2 CalibrationRestartCommand	81
8.17.4.3 GazePoint	82
8.17.4.4 GazeVisibilityCommand	82
8.18 Tobii.Research.Addons.CalibrationValidationPoint Class Reference	82
8.18.1 Detailed Description	83
8.18.2 Member Function Documentation	83
8.18.2.1 ToString()	83
8.18.3 Property Documentation	83
8.18.3.1 AccuracyLeftEye	83
8.18.3.2 AccuracyRightEye	83
8.18.3.3 Coordinates	83
8.18.3.4 GazeData	84
8.18.3.5 PrecisionLeftEye	84
8.18.3.6 PrecisionRightEye	84
8.18.3.7 PrecisionRMSLeftEye	84
8.18.3.8 PrecisionRMSRightEye	84
8.18.3.9 TimedOut	84
8.19 Tobii.Research.Addons.CalibrationValidationResult Class Reference	85
8.19.1 Detailed Description	85
8.19.2 Member Function Documentation	85
8.19.2.1 ToString()	85
8.19.3 Property Documentation	86
8.19.3.1 AverageAccuracyLeftEye	86
8.19.3.2 AverageAccuracyRightEye	86
8.19.3.3 AveragePrecisionLeftEye	86
8.19.3.4 AveragePrecisionRightEye	86
8.19.3.5 AveragePrecisionRMSLeftEye	86
8.19.3.6 AveragePrecisionRMSRightEye	87
8.19.3.7 Points	87
8.20 CustomCalibrationLibrary.ViewModels.CalibrationViewModel Class Reference	87



8.20.1 Detailed Description . . . . .	88
8.20.2 Constructor & Destructor Documentation . . . . .	88
8.20.2.1 CalibrationViewModel() . . . . .	88
8.20.3 Member Data Documentation . . . . .	89
8.20.3.1 _model . . . . .	89
8.20.4 Property Documentation . . . . .	89
8.20.4.1 CalibrationPoints . . . . .	89
8.21 CustomCalibrationLibrary.Views.CalibrationWindow Class Reference . . . . .	89
8.21.1 Detailed Description . . . . .	90
8.22 CustomCalibrationLibrary.Views.Computing Class Reference . . . . .	90
8.22.1 Detailed Description . . . . .	91
8.22.2 Constructor & Destructor Documentation . . . . .	91
8.22.2.1 Computing() . . . . .	91
8.23 GazeUtilityLibrary.ConfigItem Class Reference . . . . .	91
8.23.1 Detailed Description . . . . .	93
8.23.2 Constructor & Destructor Documentation . . . . .	93
8.23.2.1 ConfigItem() . . . . .	94
8.23.3 Property Documentation . . . . .	94
8.23.3.1 CalibrationLogColumnOrder . . . . .	94
8.23.3.2 CalibrationLogColumnTitle . . . . .	94
8.23.3.3 CalibrationLogWriteOutput . . . . .	94
8.23.3.4 CalibrationPoints . . . . .	94
8.23.3.5 ConfigName . . . . .	94
8.23.3.6 DataLogColumnOrder . . . . .	95
8.23.3.7 DataLogColumnTitle . . . . .	95
8.23.3.8 DataLogCount . . . . .	95
8.23.3.9 DataLogDisabledOnStartup . . . . .	95
8.23.3.10 DataLogFormatDiameter . . . . .	95
8.23.3.11 DataLogFormatNormalizedPoint . . . . .	95
8.23.3.12 DataLogFormatOrigin . . . . .	96
8.23.3.13 DataLogFormatTimeStamp . . . . .	96
8.23.3.14 DataLogFormatTimeStampRelative . . . . .	96
8.23.3.15 DataLogFormatValidation . . . . .	96
8.23.3.16 DataLogPath . . . . .	96
8.23.3.17 DataLogWriteOutput . . . . .	96
8.23.3.18 DriftCompensationDispersionThreshold . . . . .	97
8.23.3.19 DriftCompensationDispersionThresholdMax . . . . .	97
8.23.3.20 DriftCompensationDurationThreshold . . . . .	97
8.23.3.21 DriftCompensationTimer . . . . .	97
8.23.3.22 DriftCompensationWindowShow . . . . .	97
8.23.3.23 LicensePath . . . . .	97
8.23.3.24 MouseCalibrationHide . . . . .	98

8.23.3.25 MouseControl . . . . .	98
8.23.3.26 MouseControlHide . . . . .	98
8.23.3.27 MouseStandardIconPath . . . . .	98
8.23.3.28 ReadyTimer . . . . .	98
8.23.3.29 ScreenArea . . . . .	98
8.23.3.30 TobiiApplicationPath . . . . .	99
8.23.3.31 TobiiCalibrate . . . . .	99
8.23.3.32 TobiiCalibrateArguments . . . . .	99
8.23.3.33 TrackerDevice . . . . .	99
8.23.3.34 ValidationDurationThreshold . . . . .	99
8.23.3.35 ValidationLogColumnOrder . . . . .	99
8.23.3.36 ValidationLogColumnTitle . . . . .	100
8.23.3.37 ValidationLogWriteOutput . . . . .	100
8.23.3.38 ValidationPoints . . . . .	100
8.23.3.39 ValidationTimer . . . . .	100
8.24 GazeUtilityLibrary.ConfigScreenArea Class Reference . . . . .	100
8.24.1 Detailed Description . . . . .	101
8.24.2 Constructor & Destructor Documentation . . . . .	101
8.24.2.1 ConfigScreenArea() [1/2] . . . . .	101
8.24.2.2 ConfigScreenArea() [2/2] . . . . .	101
8.24.3 Property Documentation . . . . .	102
8.24.3.1 BottomLeft . . . . .	102
8.24.3.2 BottomRight . . . . .	102
8.24.3.3 Center . . . . .	102
8.24.3.4 Height . . . . .	102
8.24.3.5 TopLeft . . . . .	102
8.24.3.6 TopRight . . . . .	103
8.24.3.7 Width . . . . .	103
8.25 CustomCalibrationLibrary.Views.Disconnect Class Reference . . . . .	103
8.25.1 Detailed Description . . . . .	104
8.25.2 Constructor & Destructor Documentation . . . . .	104
8.25.2.1 Disconnect() . . . . .	104
8.25.3 Property Documentation . . . . .	104
8.25.3.1 CalibrationAbortCommand . . . . .	104
8.26 GazeUtilityLibrary.DriftCompensation Class Reference . . . . .	105
8.26.1 Detailed Description . . . . .	105
8.26.2 Constructor & Destructor Documentation . . . . .	105
8.26.2.1 DriftCompensation() . . . . .	105
8.26.3 Member Function Documentation . . . . .	106
8.26.3.1 Reset() . . . . .	106
8.26.3.2 Start() . . . . .	106
8.26.3.3 Update() . . . . .	106

8.26.4 Property Documentation . . . . .	106
8.26.4.1 DeviationAngle . . . . .	106
8.26.4.2 Dispersion . . . . .	107
8.26.4.3 Q . . . . .	107
8.27 GazeUtilityLibrary.DataStructs.DriftCompensationData Class Reference . . . . .	107
8.27.1 Detailed Description . . . . .	107
8.27.2 Constructor & Destructor Documentation . . . . .	107
8.27.2.1 DriftCompensationData() . . . . .	107
8.27.3 Property Documentation . . . . .	108
8.27.3.1 Compensation . . . . .	108
8.27.3.2 GazePosition2d . . . . .	108
8.27.3.3 GazePosition3d . . . . .	108
8.28 CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel Class Reference . . . . .	108
8.28.1 Detailed Description . . . . .	109
8.28.2 Constructor & Destructor Documentation . . . . .	109
8.28.2.1 DriftCompensationViewModel() . . . . .	109
8.28.3 Property Documentation . . . . .	109
8.28.3.1 FixationPoint . . . . .	109
8.29 CustomCalibrationLibrary.Views.DriftCompensationWindow Class Reference . . . . .	110
8.29.1 Detailed Description . . . . .	110
8.29.2 Constructor & Destructor Documentation . . . . .	110
8.29.2.1 DriftCompensationWindow() . . . . .	111
8.30 GazeUtilityLibrary.DataStructs.EyeData Class Reference . . . . .	111
8.30.1 Detailed Description . . . . .	111
8.30.2 Constructor & Destructor Documentation . . . . .	111
8.30.2.1 EyeData() . . . . .	111
8.30.3 Property Documentation . . . . .	112
8.30.3.1 IsPupilDiameterValid . . . . .	112
8.30.3.2 PupilDiameter . . . . .	112
8.31 GazeUtilityLibrary.Tracker.EyeTrackerPro Class Reference . . . . .	112
8.31.1 Detailed Description . . . . .	114
8.31.2 Constructor & Destructor Documentation . . . . .	114
8.31.2.1 EyeTrackerPro() . . . . .	114
8.31.3 Member Function Documentation . . . . .	114
8.31.3.1 ApplyCalibration() . . . . .	114
8.31.3.2 CollectCalibrationDataAsync() . . . . .	115
8.31.3.3 CollectValidationDataAsync() . . . . .	115
8.31.3.4 ComputeValidation() . . . . .	116
8.31.3.5 FinishCalibration() . . . . .	116
8.31.3.6 FinishCalibrationAsync() . . . . .	116
8.31.3.7 FinishValidation() . . . . .	116
8.31.3.8 GetFixationFrameCount() . . . . .	116

8.31.3.9 GetUnitDirection()	117
8.31.3.10 InitCalibration()	117
8.31.3.11 InitCalibrationAsync()	117
8.31.3.12 InitDriftCompensation()	118
8.31.3.13 InitValidation()	118
8.31.3.14 IsInitialised()	118
8.31.3.15 IsLicenseOk()	118
8.31.3.16 PatternReplace()	119
8.32 CustomCalibrationLibrary.Views.FixationPoint Class Reference	119
8.32.1 Detailed Description	120
8.32.2 Constructor & Destructor Documentation	120
8.32.2.1 FixationPoint()	120
8.33 GazeUtilityLibrary.DataStructs.GazeCalibrationData Class Reference	120
8.33.1 Detailed Description	121
8.33.2 Constructor & Destructor Documentation	121
8.33.2.1 GazeCalibrationData()	121
8.33.3 Member Function Documentation	122
8.33.3.1 Prepare()	122
8.33.4 Property Documentation	122
8.33.4.1 ValidityLeft	122
8.33.4.2 ValidityRight	122
8.33.4.3 XCoord	122
8.33.4.4 XCoordLeft	123
8.33.4.5 XCoordRight	123
8.33.4.6 YCoord	123
8.33.4.7 YCoordLeft	123
8.33.4.8 YCoordRight	123
8.34 GazeUtilityLibrary.GazeConfigError Class Reference	124
8.34.1 Detailed Description	125
8.34.2 Member Function Documentation	125
8.34.2.1 GetGazeConfigErrorString()	125
8.34.3 Property Documentation	125
8.34.3.1 Error	125
8.35 GazeUtilityLibrary.GazeConfiguration Class Reference	125
8.35.1 Detailed Description	126
8.35.2 Constructor & Destructor Documentation	126
8.35.2.1 GazeConfiguration()	126
8.35.3 Member Function Documentation	127
8.35.3.1 CleanupCalibrationOutputFile()	127
8.35.3.2 CleanupGazeOutputFile()	127
8.35.3.3 CleanupValidationOutputFile()	127
8.35.3.4 DumpCurrentConfigurationFile()	128

8.35.3.5 InitConfig()	128
8.35.3.6 PrepareCalibrationOutputFile()	128
8.35.3.7 PrepareGazeOutputFile()	129
8.35.3.8 PrepareValidationOutputFile()	129
8.35.3.9 WriteToCalibrationOutput()	129
8.35.3.10 WriteToGazeOutput()	130
8.35.3.11 WriteToValidationOutput()	130
8.35.4 Property Documentation	130
8.35.4.1 Config	130
8.36 GazeUtilityLibrary.DataStructs.GazeData Class Reference	131
8.36.1 Detailed Description	131
8.36.2 Constructor & Destructor Documentation	131
8.36.2.1 GazeData() [1/3]	131
8.36.2.2 GazeData() [2/3]	132
8.36.2.3 GazeData() [3/3]	132
8.36.3 Member Function Documentation	133
8.36.3.1 Prepare()	133
8.36.4 Property Documentation	134
8.36.4.1 Combined	134
8.36.4.2 DriftCompensation	134
8.36.4.3 Left	134
8.36.4.4 Right	134
8.36.4.5 Timestamp	134
8.36.4.6 TimestampReceived	135
8.37 GazeUtilityLibrary.DataStructs.GazeData2d Class Reference	135
8.37.1 Detailed Description	135
8.37.2 Constructor & Destructor Documentation	135
8.37.2.1 GazeData2d()	135
8.37.3 Property Documentation	136
8.37.3.1 GazePoint	136
8.37.3.2 IsGazePointValid	136
8.38 GazeUtilityLibrary.DataStructs.GazeData3d Class Reference	136
8.38.1 Detailed Description	137
8.38.2 Constructor & Destructor Documentation	137
8.38.2.1 GazeData3d()	137
8.38.3 Property Documentation	137
8.38.3.1 GazeDirection	137
8.38.3.2 GazeDistance	137
8.38.3.3 GazeOrigin	138
8.38.3.4 GazePoint	138
8.38.3.5 IsGazeOriginValid	138
8.38.3.6 IsGazePointValid	138

8.39 GazeUtilityLibrary.DataStructs.GazeDataCollection Class Reference . . . . .	138
8.39.1 Detailed Description . . . . .	139
8.39.2 Constructor & Destructor Documentation . . . . .	139
8.39.2.1 GazeDataCollection() [1/2] . . . . .	139
8.39.2.2 GazeDataCollection() [2/2] . . . . .	139
8.39.3 Property Documentation . . . . .	140
8.39.3.1 EyeData . . . . .	140
8.39.3.2 GazeData2d . . . . .	140
8.39.3.3 GazeData3d . . . . .	140
8.40 GazeUtilityLibrary.GazeDataError Class Reference . . . . .	141
8.40.1 Detailed Description . . . . .	142
8.40.2 Member Function Documentation . . . . .	142
8.40.2.1 GetGazeDataErrorString() . . . . .	142
8.40.3 Property Documentation . . . . .	142
8.40.3.1 Error . . . . .	142
8.41 GazeUtilityLibrary.GazeError Class Reference . . . . .	142
8.41.1 Detailed Description . . . . .	143
8.41.2 Member Function Documentation . . . . .	143
8.41.2.1 ConvertToBinString() . . . . .	143
8.42 GazeUtilityLibrary.DataStructs.GazeValidationData Class Reference . . . . .	143
8.42.1 Detailed Description . . . . .	144
8.42.2 Constructor & Destructor Documentation . . . . .	144
8.42.2.1 GazeValidationData() [1/2] . . . . .	144
8.42.2.2 GazeValidationData() [2/2] . . . . .	144
8.42.3 Member Function Documentation . . . . .	145
8.42.3.1 AddPoint() . . . . .	145
8.42.4 Property Documentation . . . . .	145
8.42.4.1 AccuracyLeft . . . . .	146
8.42.4.2 AccuracyRight . . . . .	146
8.42.4.3 Points . . . . .	146
8.42.4.4 PrecisionLeft . . . . .	146
8.42.4.5 PrecisionRight . . . . .	146
8.42.4.6 PrecisionRmsLeft . . . . .	146
8.42.4.7 PrecisionRmsRight . . . . .	147
8.43 GazeUtilityLibrary.DataStructs.GazeValidationPoint Class Reference . . . . .	147
8.43.1 Detailed Description . . . . .	147
8.43.2 Constructor & Destructor Documentation . . . . .	147
8.43.2.1 GazeValidationPoint() . . . . .	147
8.43.3 Member Function Documentation . . . . .	148
8.43.3.1 Prepare() . . . . .	148
8.43.4 Property Documentation . . . . .	148
8.43.4.1 Point . . . . .	148

8.43.4.2 Result	148
8.44 CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter Class Reference	149
8.44.1 Detailed Description	149
8.44.2 Member Function Documentation	150
8.44.2.1 Convert()	150
8.44.2.2 ConvertBack()	150
8.45 GazeUtilityLibrary.JsonConfigParser Class Reference	151
8.45.1 Detailed Description	151
8.45.2 Constructor & Destructor Documentation	151
8.45.2.1 JsonConfigParser()	151
8.45.3 Member Function Documentation	152
8.45.3.1 GetDefaultConfig()	152
8.45.3.2 ParseJsonConfig()	152
8.45.3.3 SerializeJsonConfig()	152
8.46 GazeUtilityLibrary.DataStructs.LiveGazePoint Class Reference	153
8.46.1 Detailed Description	154
8.46.2 Property Documentation	154
8.46.2.1 Visibility	154
8.46.2.2 X	154
8.46.2.3 Y	154
8.46.3 Event Documentation	154
8.46.3.1 PropertyChanged	154
8.47 CustomCalibrationLibrary.ViewModels.Monitor Class Reference	155
8.47.1 Detailed Description	155
8.47.2 Constructor & Destructor Documentation	155
8.47.2.1 Monitor()	155
8.47.3 Property Documentation	155
8.47.3.1 Index	155
8.47.3.2 Name	156
8.48 GazeUtilityLibrary.MouseHider Class Reference	156
8.48.1 Detailed Description	156
8.48.2 Constructor & Destructor Documentation	156
8.48.2.1 MouseHider()	156
8.48.3 Member Function Documentation	157
8.48.3.1 HideCursor()	157
8.48.3.2 ShowCursor()	157
8.49 GazeUtilityLibrary.Tracker.MouseTracker Class Reference	157
8.49.1 Detailed Description	159
8.49.2 Constructor & Destructor Documentation	159
8.49.2.1 MouseTracker()	159
8.49.3 Member Function Documentation	160
8.49.3.1 ApplyCalibration()	160

8.49.3.2 CollectCalibrationDataAsync()	160
8.49.3.3 CollectValidationDataAsync()	160
8.49.3.4 ComputeValidation()	161
8.49.3.5 Dispose()	161
8.49.3.6 FinishCalibration()	161
8.49.3.7 FinishCalibrationAsync()	162
8.49.3.8 FinishValidation()	162
8.49.3.9 GetFixationFrameCount()	162
8.49.3.10 GetUnitDirection()	162
8.49.3.11 InitCalibration()	163
8.49.3.12 InitCalibrationAsync()	163
8.49.3.13 InitDriftCompensation()	163
8.49.3.14 InitValidation()	163
8.49.3.15 Start()	164
8.49.3.16 Stop()	164
8.50 GazeUtilityLibrary.DataStructs.PipeCommand Class Reference	164
8.50.1 Detailed Description	164
8.50.2 Constructor & Destructor Documentation	164
8.50.2.1 PipeCommand()	164
8.50.3 Property Documentation	165
8.50.3.1 Command	165
8.50.3.2 Label	165
8.50.3.3 ResetStartTime	165
8.50.3.4 TrialId	165
8.51 CustomCalibrationLibrary.Converters.PositionConverter Class Reference	166
8.51.1 Detailed Description	167
8.51.2 Member Function Documentation	167
8.51.2.1 Convert()	167
8.51.2.2 ConvertBack()	167
8.51.3 Member Data Documentation	168
8.51.3.1 OffsetProperty	168
8.51.4 Property Documentation	168
8.51.4.1 Offset	168
8.52 CustomCalibrationLibrary.Converters.ProximityColorConverter Class Reference	169
8.52.1 Detailed Description	169
8.52.2 Member Function Documentation	170
8.52.2.1 Convert()	170
8.52.2.2 ConvertBack()	170
8.53 GazeUtilityLibrary.ScreenArea Class Reference	171
8.53.1 Detailed Description	172
8.53.2 Constructor & Destructor Documentation	172
8.53.2.1 ScreenArea()	172



8.53.3 Member Function Documentation	172
8.53.3.1 Dump()	172
8.53.3.2 GetIntersectionPoint()	173
8.53.3.3 GetPoint2d()	173
8.53.3.4 GetPoint2dNormalized()	173
8.53.4 Property Documentation	175
8.53.4.1 BottomLeft	175
8.53.4.2 BottomRight	175
8.53.4.3 Center	175
8.53.4.4 Height	175
8.53.4.5 TopLeft	176
8.53.4.6 TopRight	176
8.53.4.7 Width	176
8.54 Tobii.Research.Addons.ScreenBasedCalibrationValidation Class Reference	176
8.54.1 Detailed Description	178
8.54.2 Member Enumeration Documentation	178
8.54.2.1 ValidationState	178
8.54.3 Constructor & Destructor Documentation	178
8.54.3.1 ScreenBasedCalibrationValidation()	178
8.54.4 Member Function Documentation	179
8.54.4.1 Compute()	179
8.54.4.2 DiscardData()	179
8.54.4.3 Dispose()	179
8.54.4.4 EnterValidationMode()	179
8.54.4.5 LeaveValidationMode()	180
8.54.4.6 StartCollectingData()	180
8.54.4.7 ToString()	180
8.54.5 Property Documentation	180
8.54.5.1 Result	180
8.54.5.2 State	181
8.55 CustomCalibrationLibrary.Views.ScreenSelection Class Reference	181
8.55.1 Detailed Description	182
8.55.2 Constructor & Destructor Documentation	182
8.55.2.1 ScreenSelection()	182
8.56 CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel Class Reference	182
8.56.1 Detailed Description	183
8.56.2 Constructor & Destructor Documentation	183
8.56.2.1 ScreenSelectionViewModel()	183
8.56.3 Property Documentation	183
8.56.3.1 CalibrationAbortCommand	183
8.56.3.2 CalibrationStartCommand	183
8.56.3.3 Monitors	184

8.56.3.4 ScreenSwitchCommand . . . . .	184
8.57 GazeUtilityLibrary.ScreenTriangle Class Reference . . . . .	184
8.57.1 Detailed Description . . . . .	185
8.57.2 Constructor & Destructor Documentation . . . . .	185
8.57.2.1 ScreenTriangle() . . . . .	185
8.57.3 Member Function Documentation . . . . .	185
8.57.3.1 GetIntersectionPoint() . . . . .	185
8.57.4 Property Documentation . . . . .	186
8.57.4.1 E1 . . . . .	186
8.57.4.2 E2 . . . . .	186
8.57.4.3 V1 . . . . .	186
8.57.4.4 V2 . . . . .	186
8.57.4.5 V3 . . . . .	186
8.58 GazeUtilityLibrary.TrackerLogger Class Reference . . . . .	187
8.58.1 Detailed Description . . . . .	187
8.58.2 Constructor & Destructor Documentation . . . . .	187
8.58.2.1 TrackerLogger() . . . . .	187
8.58.3 Member Function Documentation . . . . .	187
8.58.3.1 Debug() . . . . .	187
8.58.3.2 DumpFatal() . . . . .	188
8.58.3.3 Error() . . . . .	188
8.58.3.4 Info() . . . . .	188
8.58.3.5 Warning() . . . . .	189
8.59 GazeUtilityLibrary.TrackerMessageBox Class Reference . . . . .	189
8.59.1 Detailed Description . . . . .	190
8.60 GazeUtilityLibrary.DataStructs.UserPositionData Class Reference . . . . .	190
8.60.1 Detailed Description . . . . .	191
8.60.2 Constructor & Destructor Documentation . . . . .	191
8.60.2.1 UserPositionData() [1/2] . . . . .	191
8.60.2.2 UserPositionData() [2/2] . . . . .	191
8.60.3 Property Documentation . . . . .	192
8.60.3.1 XCoordLeft . . . . .	192
8.60.3.2 XCoordRight . . . . .	192
8.60.3.3 YCoordLeft . . . . .	192
8.60.3.4 YCoordRight . . . . .	192
8.60.3.5 ZCoordLeft . . . . .	193
8.60.3.6 ZCoordRight . . . . .	193
8.60.4 Event Documentation . . . . .	193
8.60.4.1 PropertyChanged . . . . .	193
8.61 CustomCalibrationLibrary.Views.UserPositionGuide Class Reference . . . . .	193
8.61.1 Detailed Description . . . . .	194
8.61.2 Constructor & Destructor Documentation . . . . .	194

8.61.2.1 UserPositionGuide()	194
8.62 CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel Class Reference	195
8.62.1 Detailed Description	195
8.62.2 Constructor & Destructor Documentation	195
8.62.2.1 UserPositionGuideViewModel()	195
8.62.3 Property Documentation	195
8.62.3.1 CalibrationAbortCommand	196
8.62.3.2 CalibrationStartCommand	196
8.62.3.3 UserPosition	196
8.63 CustomCalibrationLibrary.Views.ValidationResult Class Reference	196
8.63.1 Detailed Description	197
8.63.2 Constructor & Destructor Documentation	197
8.63.2.1 ValidationResult()	197
8.64 CustomCalibrationLibrary.ViewModels.ValidationResultViewModel Class Reference	198
8.64.1 Detailed Description	198
8.64.2 Constructor & Destructor Documentation	198
8.64.2.1 ValidationResultViewModel()	198
8.64.3 Property Documentation	198
8.64.3.1 ValidationCloseCommand	199
8.64.3.2 ValidationData	199
8.64.3.3 ValidationRestartCommand	199
<b>Index</b>	<b>201</b>



# Chapter 1

## Changelog

### v3.4.0

#### New Features

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

#### Improvements

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

## Changes

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

## Bug Fixes

- Fix dispersion computation during drift compensation.

## v3.3.2

### New Features

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

### Improvements

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

### Bug Fixes

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

## v3.3.1

### Improvements

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

### Bug Fixes

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

---

## v3.3.0

### New Features

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

### Improvements

- Fix compiler warnings.
- Extend helper scripts.

## v3.2.0

### New Features

- Add relative timestamp to output data.
- Add annotation tag to 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\*\***. **GazeControl.exe** accepts the following optional arguments:
  - `/reset`: Allows to reset the relative timestamp of the gaze data.
  - `/trialId <ID>`: Sets a trial ID `<ID>` which will be added to each data sample in the output file. **Important:** Make sure that only integer numbers are used as trial ID.
  - `/label <LABEL>`: Sets a custom label `<LABEL>` which will be added to each data sample in the output file. Any string is accepted here.
  - `/command <COMMAND>`: A command allow to activate/deactivate features of `Gaze.exe`. The following commands are supported:
    - \* `CUSTOM_CALIBRATE` uses the [Tobii Pro SDK](#) and launches a custom calibration process which allows to calibrate the eye tracker without having to rely on the calibration software provided by [Tobii](#).

- \* `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 `DRIFT_COMPENSATION`.
- \* `TERMINATE` requests `**Gaze.exe**` to close gracefully and logs these events to the log file.

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

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

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

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

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

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

## Release Notes

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





## Chapter 3

# Sample Files for Experimentation with Eye Tracker Utility

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

### **config.json**

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

### **config\_libgac.json**

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

### **template.osexp**

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

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

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

### **template.ztt**

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



## Chapter 4

# Namespace Index

### 4.1 Namespace List

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

<a href="#">CustomCalibrationLibrary</a>	23
<a href="#">CustomCalibrationLibrary.Commands</a>	23
<a href="#">CustomCalibrationLibrary.Converters</a>	23
<a href="#">CustomCalibrationLibrary.Models</a>	23
<a href="#">CustomCalibrationLibrary.ViewModels</a>	24
<a href="#">CustomCalibrationLibrary.Views</a>	25
<a href="#">GazeControl</a>	25
<a href="#">GazeToMouse</a>	25
<a href="#">GazeUtilityLibrary</a>	
helper class to show and hide the system curser	26
<a href="#">GazeUtilityLibrary.DataStructs</a>	28
<a href="#">GazeUtilityLibrary.Tracker</a>	29
<a href="#">ShowMouse</a>	30
<a href="#">Tobii</a>	30
<a href="#">Tobii.Research</a>	30
<a href="#">Tobii.Research.Addons</a>	30
<a href="#">Tobii.Research.Addons.Utility</a>	30
<a href="#">TobiiCalibrate</a>	30



## Chapter 5

# Hierarchical Index

### 5.1 Class Hierarchy

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

Application	
GazeControl.App . . . . .	31
GazeToMouse.App . . . . .	32
ShowMouse.App . . . . .	36
TobiiCalibrate.App . . . . .	37
Tobii.Research.Addons.CalibrationValidationPoint . . . . .	82
Tobii.Research.Addons.CalibrationValidationResult . . . . .	85
CustomCalibrationLibrary.ViewModels.CalibrationViewModel . . . . .	87
CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel . . . . .	79
GazeUtilityLibrary.ConfigItem . . . . .	91
GazeUtilityLibrary.ConfigScreenArea . . . . .	100
DependencyObject	
CustomCalibrationLibrary.Converters.PositionConverter . . . . .	166
GazeUtilityLibrary.DriftCompensation . . . . .	105
GazeUtilityLibrary.DataStructs.DriftCompensationData . . . . .	107
CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel . . . . .	108
GazeUtilityLibrary.DataStructs.EyeData . . . . .	111
Frame	
CustomCalibrationLibrary.Views.CalibrationFrame . . . . .	62
GazeUtilityLibrary.DataStructs.GazeCalibrationData . . . . .	120
GazeUtilityLibrary.GazeConfiguration . . . . .	125
GazeUtilityLibrary.DataStructs.GazeData . . . . .	131
GazeUtilityLibrary.DataStructs.GazeData2d . . . . .	135
GazeUtilityLibrary.DataStructs.GazeData3d . . . . .	136
GazeUtilityLibrary.DataStructs.GazeDataCollection . . . . .	138
GazeUtilityLibrary.GazeError . . . . .	142
GazeUtilityLibrary.CalibrationDataError . . . . .	57
GazeUtilityLibrary.GazeConfigError . . . . .	124
GazeUtilityLibrary.GazeDataError . . . . .	141
GazeUtilityLibrary.DataStructs.GazeValidationData . . . . .	143
GazeUtilityLibrary.DataStructs.GazeValidationPoint . . . . .	147
ICommand	
CustomCalibrationLibrary.Commands.CalibrationCommand . . . . .	55
IDisposable	
GazeUtilityLibrary.Tracker.BaseTracker . . . . .	38

GazeUtilityLibrary.Tracker.EyeTrackerPro . . . . .	112
GazeUtilityLibrary.Tracker.MouseTracker . . . . .	157
Tobii.Research.Addons.ScreenBasedCalibrationValidation . . . . .	176
INotifyPropertyChanged	
CustomCalibrationLibrary.Models.CalibrationModel . . . . .	63
CustomCalibrationLibrary.Views.CalibrationFailed . . . . .	59
GazeUtilityLibrary.DataStructs.CalibrationPoint . . . . .	70
CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel . . . . .	74
GazeUtilityLibrary.DataStructs.LiveGazePoint . . . . .	153
GazeUtilityLibrary.DataStructs.UserPositionData . . . . .	190
GazeUtilityLibrary.Tracker.BaseTracker . . . . .	38
IValueConverter	
CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter . . . . .	149
CustomCalibrationLibrary.Converters.PositionConverter . . . . .	166
CustomCalibrationLibrary.Converters.ProximityColorConverter . . . . .	169
GazeUtilityLibrary.JsonConfigParser . . . . .	151
CustomCalibrationLibrary.ViewModels.Monitor . . . . .	155
GazeUtilityLibrary.MouseHider . . . . .	156
Page	
CustomCalibrationLibrary.Views.Calibration . . . . .	53
CustomCalibrationLibrary.Views.CalibrationFailed . . . . .	59
CustomCalibrationLibrary.Views.Computing . . . . .	90
CustomCalibrationLibrary.Views.Disconnect . . . . .	103
Page	
CustomCalibrationLibrary.Views.CalibrationResult . . . . .	76
CustomCalibrationLibrary.Views.ScreenSelection . . . . .	181
CustomCalibrationLibrary.Views.UserPositionGuide . . . . .	193
CustomCalibrationLibrary.Views.ValidationResult . . . . .	196
GazeUtilityLibrary.DataStructs.PipeCommand . . . . .	164
GazeUtilityLibrary.ScreenArea . . . . .	171
CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel . . . . .	182
GazeUtilityLibrary.ScreenTriangle . . . . .	184
GazeUtilityLibrary.TrackerLogger . . . . .	187
UserControl	
CustomCalibrationLibrary.Views.CalibrationPoint . . . . .	73
CustomCalibrationLibrary.Views.CalibrationResultPoint . . . . .	78
CustomCalibrationLibrary.Views.FixationPoint . . . . .	119
CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel . . . . .	195
CustomCalibrationLibrary.ViewModels.ValidationResultViewModel . . . . .	198
Window	
CustomCalibrationLibrary.Views.CalibrationWindow . . . . .	89
CustomCalibrationLibrary.Views.DriftCompensationWindow . . . . .	110
GazeUtilityLibrary.TrackerMessageBox . . . . .	189

## Chapter 6

# Class Index

### 6.1 Class List

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

<a href="#">GazeControl.App</a>	
Interaction logic for App.xaml . . . . .	31
<a href="#">GazeToMouse.App</a>	
Interaction logic for App.xaml . . . . .	32
<a href="#">ShowMouse.App</a>	
Interaction logic for App.xaml . . . . .	36
<a href="#">TobiiCalibrate.App</a>	
Interaction logic for App.xaml . . . . .	37
<a href="#">GazeUtilityLibrary.Tracker.BaseTracker</a>	
The common interface for the <a href="#">Tobii</a> eyetracker Engines Core and Pro . . . . .	38
<a href="#">CustomCalibrationLibrary.Views.Calibration</a>	
Interaction logic for Calibration.xaml . . . . .	53
<a href="#">CustomCalibrationLibrary.Commands.CalibrationCommand</a>	
Comand class to trigger calibration events . . . . .	55
<a href="#">GazeUtilityLibrary.CalibrationDataError</a>	
The calibration data error class to convert error flags to binary strings . . . . .	57
<a href="#">CustomCalibrationLibrary.Views.CalibrationFailed</a>	
Interaction logic for CalibrationFailed.xaml . . . . .	59
<a href="#">CustomCalibrationLibrary.Views.CalibrationFrame</a>	
Interaction logic for CalibrationCollection.xaml . . . . .	62
<a href="#">CustomCalibrationLibrary.Models.CalibrationModel</a>	
The model for the calibration process . . . . .	63
<a href="#">GazeUtilityLibrary.DataStructs.CalibrationPoint</a>	
A calibration point class holding several metrics connected to a calibration point . . . . .	70
<a href="#">CustomCalibrationLibrary.Views.CalibrationPoint</a>	
Interaction logic for CalibrationPoint.xaml . . . . .	73
<a href="#">CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel</a>	
The view model for a calibration point . . . . .	74
<a href="#">CustomCalibrationLibrary.Views.CalibrationResult</a>	
Interaction logic for CalibrationResult.xaml . . . . .	76
<a href="#">CustomCalibrationLibrary.Views.CalibrationResultPoint</a>	
Interaction logic for CalibrationResultPoint.xaml . . . . .	78
<a href="#">CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel</a>	
View model class of the gaze calibration result . . . . .	79

<a href="#">Tobii.Research.Addons.CalibrationValidationPoint</a>	
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 . . . . .	82
<a href="#">Tobii.Research.Addons.CalibrationValidationResult</a>	
Contains the result of the calibration validation . . . . .	85
<a href="#">CustomCalibrationLibrary.ViewModels.CalibrationViewModel</a>	
The view model class of the calibration view . . . . .	87
<a href="#">CustomCalibrationLibrary.Views.CalibrationWindow</a>	
Interaction logic for MainWindow.xaml . . . . .	89
<a href="#">CustomCalibrationLibrary.Views.Computing</a>	
Interaction logic for Computing.xaml . . . . .	90
<a href="#">GazeUtilityLibrary.ConfigItem</a>	
configuration file class . . . . .	91
<a href="#">GazeUtilityLibrary.ConfigScreenArea</a>	
The JSON structure of the screen area . . . . .	100
<a href="#">CustomCalibrationLibrary.Views.Disconnect</a>	
Interaction logic for Disconnect.xaml . . . . .	103
<a href="#">GazeUtilityLibrary.DriftCompensation</a>	
The class to handle drift compensation . . . . .	105
<a href="#">GazeUtilityLibrary.DataStructs.DriftCompensationData</a>	
The drift compensation data structure . . . . .	107
<a href="#">CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel</a>	
The view model class of the drift compensation view . . . . .	108
<a href="#">CustomCalibrationLibrary.Views.DriftCompensationWindow</a>	
Interaction logic for DriftCompensation.xaml . . . . .	110
<a href="#">GazeUtilityLibrary.DataStructs.EyeData</a>	
The eye data set, including pupil information . . . . .	111
<a href="#">GazeUtilityLibrary.Tracker.EyeTrackerPro</a>	
Interface to the <a href="#">Tobii</a> SDK Pro engine . . . . .	112
<a href="#">CustomCalibrationLibrary.Views.FixationPoint</a>	
Interaction logic for FixationPoint.xaml . . . . .	119
<a href="#">GazeUtilityLibrary.DataStructs.GazeCalibrationData</a>	
The gaze calibration data structure . . . . .	120
<a href="#">GazeUtilityLibrary.GazeConfigError</a>	
The gaze config error class to convert error flags to binary strings . . . . .	124
<a href="#">GazeUtilityLibrary.GazeConfiguration</a>	
The gaze configuration handler . . . . .	125
<a href="#">GazeUtilityLibrary.DataStructs.GazeData</a>	
The class definition of a gaze data set . . . . .	131
<a href="#">GazeUtilityLibrary.DataStructs.GazeData2d</a>	
The 2d gaze data set . . . . .	135
<a href="#">GazeUtilityLibrary.DataStructs.GazeData3d</a>	
The 3d gaze data set . . . . .	136
<a href="#">GazeUtilityLibrary.DataStructs.GazeDataCollection</a>	
The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data . . .	138
<a href="#">GazeUtilityLibrary.GazeDataError</a>	
The gaze data error class to convert error flags to binary strings . . . . .	141
<a href="#">GazeUtilityLibrary.GazeError</a>	
The base error class to convert error flags to binary strings . . . . .	142
<a href="#">GazeUtilityLibrary.DataStructs.GazeValidationData</a>	
The gaze validation data structure . . . . .	143
<a href="#">GazeUtilityLibrary.DataStructs.GazeValidationPoint</a>	
A validation point . . . . .	147
<a href="#">CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter</a>	
Converts True to Hidden and False to Visible . . . . .	149
<a href="#">GazeUtilityLibrary.JsonConfigParser</a>	
The config file "config.json" is parsed and its values are attributed to the <a href="#">ConfigItem</a> class . . .	151



<a href="#">GazeUtilityLibrary.DataStructs.LiveGazePoint</a>	
The live gaze point used for verification during the calibration process . . . . .	153
<a href="#">CustomCalibrationLibrary.ViewModels.Monitor</a>	
A representation of the screen . . . . .	155
<a href="#">GazeUtilityLibrary.MouseHider</a>	
hide standard mouse pointer and restore it . . . . .	156
<a href="#">GazeUtilityLibrary.Tracker.MouseTracker</a>	
This class is used to hook into the system mouse events and track the position . . . . .	157
<a href="#">GazeUtilityLibrary.DataStructs.PipeCommand</a>	
The JSON structure of a pipe command . . . . .	164
<a href="#">CustomCalibrationLibrary.Converters.PositionConverter</a>	
Converter class to convert a normalized coordinate to a pixel coordinate . . . . .	166
<a href="#">CustomCalibrationLibrary.Converters.ProximityColorConverter</a>	
Converter class to convert the proximity of a normalized coordinate to the center point (0.5) into colors . . . . .	169
<a href="#">GazeUtilityLibrary.ScreenArea</a>	
The class describing the Screen area in 3d and 2d space . . . . .	171
<a href="#">Tobii.Research.Addons.ScreenBasedCalibrationValidation</a>	
Provides methods and properties for managing calibration validation for screen based eye trackers . . . . .	176
<a href="#">CustomCalibrationLibrary.Views.ScreenSelection</a>	
Interaction logic for ScreenSelection.xaml . . . . .	181
<a href="#">CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel</a>	
The view model class for the screen selection view . . . . .	182
<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 whether the gaze point is outside the screen area . . . . .	184
<a href="#">GazeUtilityLibrary.TrackerLogger</a>	
Simple logger class . . . . .	187
<a href="#">GazeUtilityLibrary.TrackerMessageBox</a>	
Interaction logic for TrackerMessageBox.xaml . . . . .	189
<a href="#">GazeUtilityLibrary.DataStructs.UserPositionData</a>	
The user position to be rendered on the screen . . . . .	190
<a href="#">CustomCalibrationLibrary.Views.UserPositionGuide</a>	
Interaction logic for UserPositionGuide.xaml . . . . .	193
<a href="#">CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel</a>	
The view model class for the user position guide view . . . . .	195
<a href="#">CustomCalibrationLibrary.Views.ValidationResult</a>	
Interaction logic for ValidationResult.xaml . . . . .	196
<a href="#">CustomCalibrationLibrary.ViewModels.ValidationResultViewModel</a>	
View model class of the gaze validation result . . . . .	198



## Chapter 7

# Namespace Documentation

### 7.1 CustomCalibrationLibrary Namespace Reference

### 7.2 CustomCalibrationLibrary.Commands Namespace Reference

#### Classes

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

### 7.3 CustomCalibrationLibrary.Converters Namespace Reference

#### Classes

- class [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 normailized coordinate to the center point (0.5) into colors.*

### 7.4 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.*

### 7.4.1 Enumeration Type Documentation

#### 7.4.1.1 CalibrationEventType

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

Events to trigger changes in the calibration process.

#### 7.4.1.2 CalibrationStatus

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

The status of the calibarion process.

## 7.5 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](#)  
    *A representation of the screen.*
- class [ScreenSelectionViewModel](#)  
    *The view model class for the screen selection view.*
- class [UserPositionGuideViewModel](#)  
    *The view model class for the user position guide view.*
- class [ValidationResultViewModel](#)  
    *View model class of the gaze validation result.*

## 7.6 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*

## 7.7 GazeControl Namespace Reference

### Classes

- class [App](#)  
*Interaction logic for App.xaml*
- class **NamedPipeClient**  
*The named pipe client handler.*

## 7.8 GazeToMouse Namespace Reference

### Classes

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

## 7.9 GazeUtilityLibrary Namespace Reference

helper class to show and hide the system cursor

### Classes

- class [CalibrationDataError](#)  
*The calibration data error class to convert error flags to binary strings.*
- class **ConfigChecker**  
*Helper class to check for the validity of configuration options.*
- class [ConfigItem](#)  
*configuration file class*
- class [ConfigScreenArea](#)  
*The JSON structure of the screen area.*
- class [DriftCompensation](#)  
*The class to handle drift compensation.*
- class [GazeConfigError](#)  
*The gaze config error class to convert error flags to binary strings.*
- class [GazeConfiguration](#)  
*The gaze configuration handler.*
- class [GazeDataError](#)  
*The gaze data error class to convert error flags to binary strings.*
- class [GazeError](#)  
*The base error class to convert error flags to binary strings.*
- class [JsonConfigParser](#)  
*The config file "config.json" is parsed and its values are attributed to the [ConfigItem](#) class.*
- class [MouseHider](#)  
*hide standard mouse pointer and 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 whether 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** }  
*A list of output files.*
- enum [EGazeConfigError](#) { **FallbackToDefaultConfigName** = 0x001, **FallbackToCurrentOutputDir** = 0x002, **FallbackToDefaultConfig** = 0x004, **FallbackToDefaultDiameterFormat** = 0x008, **FallbackToDefaultOriginFormat** = 0x010, **FallbackToDefaultTimestampFormat** = 0x020, **OmitColumnTitles** = 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*

### 7.9.1 Detailed Description

helper class to show and hide the system curser

### 7.9.2 Enumeration Type Documentation

#### 7.9.2.1 ECalibrationDataError

```
enum GazeUtilityLibrary.ECalibrationDataError [strong]
```

Error values of the gaze output data

#### 7.9.2.2 EGazeConfigError

```
enum GazeUtilityLibrary.EGazeConfigError [strong]
```

Error values of the configuration

#### 7.9.2.3 EGazeDataError

```
enum GazeUtilityLibrary.EGazeDataError [strong]
```

Error values of the gaze output data

#### 7.9.2.4 EOutputType

```
enum GazeUtilityLibrary.EOutputType [strong]
```

A list of output files.

## 7.10 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 gaze calibration data structure*
- class [GazeData](#)  
*The class definition of a gaze data set*
- class [GazeData2d](#)  
*The 2d gaze data set.*
- class [GazeData3d](#)  
*The 3d gaze data set.*
- class [GazeDataCollection](#)  
*The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data.*
- class [GazeDataConverter](#)  
*Convert values to strings according to a format.*
- class [GazeValidationData](#)  
*The gaze validation data structure*
- class [GazeValidationPoint](#)  
*A validation point.*
- class [LiveGazePoint](#)  
*The live gaze point used for verification during the calibration process.*
- class [PipeCommand](#)  
*The JSON structure of a pipe command.*
- class [UserPositionData](#)  
*The user position to be rendered on the screen.*

### Enumerations

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



- 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](#) {  
**Point2dX, Point2dY, LeftAccuracy, LeftPrecision,**  
**LeftPrecisionRMS, RightAccuracy, RightPrecision, RightPrecisionRMS }**
- enumerates output values produced by the eyetracker*

## 7.10.1 Enumeration Type Documentation

### 7.10.1.1 CalibrationOutputValue

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

enumerates output values produced by the eyetracker

### 7.10.1.2 GazeOutputValue

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

enumerates output values produced by the eyetracker

### 7.10.1.3 ValidationOutputValue

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

enumerates output values produced by the eyetracker

## 7.11 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*

## 7.12 ShowMouse Namespace Reference

### Classes

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

## 7.13 Tobii Namespace Reference

## 7.14 Tobii.Research Namespace Reference

## 7.15 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.*

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

### Classes

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

## 7.17 TobiiCalibrate Namespace Reference

### Classes

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

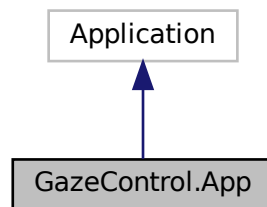
## Chapter 8

# Class Documentation

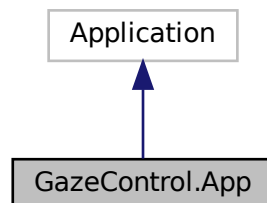
### 8.1 GazeControl.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for GazeControl.App:



Collaboration diagram for GazeControl.App:



### 8.1.1 Detailed Description

Interaction logic for App.xaml

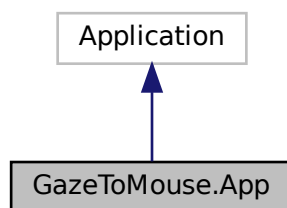
The documentation for this class was generated from the following file:

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

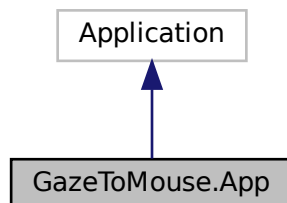
## 8.2 GazeToMouse.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for GazeToMouse.App:



Collaboration diagram for GazeToMouse.App:



## Public Member Functions

- void [GazeRecordingEnable](#) ()  
*Enable gaze recordings to disk.*
- void [GazeRecordingDisable](#) ()  
*Disable gaze recordings.*
- void [MouseTrackingEnable](#) ()  
*Enable mouse tracking which updates the mouse position to the current gaze point.*
- void [MouseTrackingDisable](#) ()  
*Disable mouse tracking.*
- void [ResetDriftCompensation](#) ()  
*Reset the current drift compensation offset to zero.*
- 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]  
*The start time of the application.*
- string [LastTag](#) [get, set]  
*The last tag to annotate gaze data.*
- string [Tag](#) [get, set]  
*An arbitrary tag to annotate gaze data.*
- int [TrialId](#) [get, set]  
*The trial ID to annotate gaze data.*

### 8.2.1 Detailed Description

Interaction logic for App.xaml

### 8.2.2 Constructor & Destructor Documentation

#### 8.2.2.1 App()

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

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

## 8.2.3 Member Function Documentation

### 8.2.3.1 CalibrationValidate()

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

Start the gaze calibration process

#### Returns

True on success, false on failure

### 8.2.3.2 CompensateDrift()

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

Start the drift compensation process

#### Returns

True on success, false on failure

### 8.2.3.3 CustomCalibrate()

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

Start the gaze calibration process

#### Returns

True on success, false on failure

### 8.2.3.4 GazeRecordingDisable()

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

Disable gaze recordings.

### 8.2.3.5 GazeRecordingEnable()

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

Enable gaze recordings to disk.

### 8.2.3.6 MouseTrackingDisable()

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

Disable mouse tracking.

### 8.2.3.7 MouseTrackingEnable()

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

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

### 8.2.3.8 ResetDriftCompensation()

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

Reset the current drift compensation offset to zero.

## 8.2.4 Property Documentation

### 8.2.4.1 LastTag

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

The last tag to annotate gaze data.

### 8.2.4.2 StartTime

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

The start time of the application.

#### 8.2.4.3 Tag

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

An arbitrary tag to annotate gaze data.

#### 8.2.4.4 TrialId

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

The trial ID to annotate gaze data.

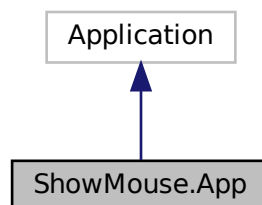
The documentation for this class was generated from the following file:

- source/GazeToMouse/App.xaml.cs

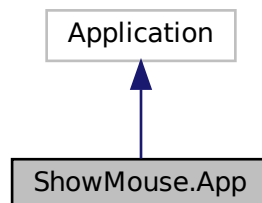
### 8.3 ShowMouse.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for ShowMouse.App:



Collaboration diagram for ShowMouse.App:





### 8.3.1 Detailed Description

Interaction logic for App.xaml

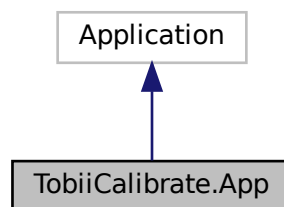
The documentation for this class was generated from the following file:

- source/ShowMouse/App.xaml.cs

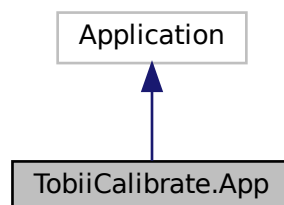
## 8.4 TobiiCalibrate.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for TobiiCalibrate.App:



Collaboration diagram for TobiiCalibrate.App:



### 8.4.1 Detailed Description

Interaction logic for App.xaml

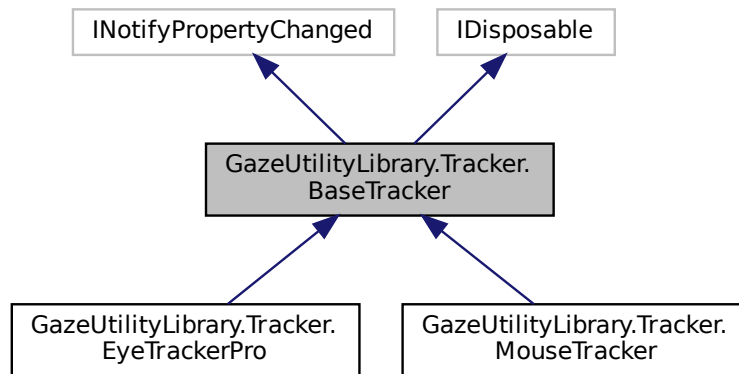
The documentation for this class was generated from the following file:

- source/TobiiCalibrate/App.xaml.cs

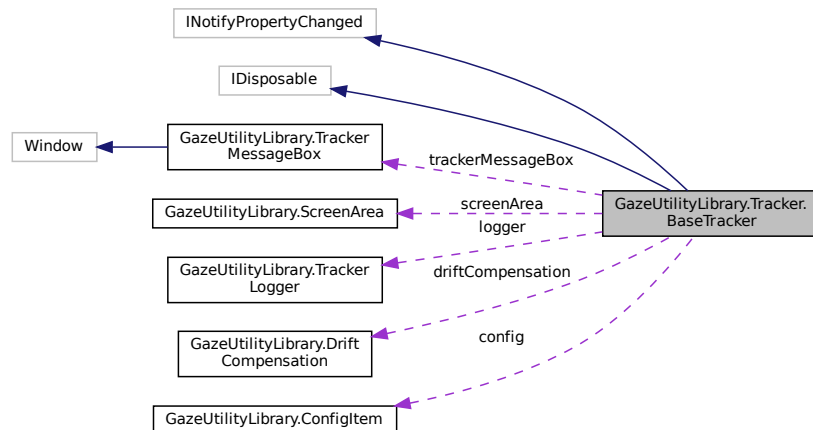
## 8.5 GazeUtilityLibrary.Tracker.BaseTracker Class Reference

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

Inheritance diagram for GazeUtilityLibrary.Tracker.BaseTracker:



Collaboration diagram for GazeUtilityLibrary.Tracker.BaseTracker:



### Public Types

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

*The tracker device status*

## Public Member Functions

- delegate void [GazeDataHandler](#) (object sender, [GazeData](#) gazeData)  
*Event handler for gaze data events of the eyetracker*
- delegate void [DriftCompensationEventHandler](#) (object sender, Quaternion [driftCompensation](#))  
*Event handler for drift compensation events*
- delegate void [UserPositionDataHandler](#) (object sender, [UserPositionData](#) e)  
*Event handler for user position data events of the eyetracker*
- [BaseTracker](#) ([TrackerLogger](#) logger, [ConfigItem](#) config, string deviceName)  
*Initializes a new instance of the EyeTrackerHandler class.*
- void [Dispose](#) ()  
*Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.*
- virtual string [PatternReplace](#) (string pattern)  
*Replaces a patten string with information from the eye tracker. This is device specific and may be overwritten by the device class.*
- abstract Task [InitCalibrationAsync](#) ()  
*Initialise the async calibartion process. This is device specific and must be overwritten by the device class.*
- abstract void [InitCalibration](#) ()  
*Initialise the calibartion process. This is device specific and must be overwritten by the device class.*
- abstract void [InitValidation](#) ()  
*Initialise the validation process. This is device specific and must be overwritten by the device class.*
- abstract Task [FinishCalibrationAsync](#) ()  
*Finish the async calibartion process. This is device specific and must be overwritten by the device class.*
- abstract void [FinishCalibration](#) ()  
*Finish the calibartion process. This is device specific and must be overwritten by the device class.*
- abstract void [FinishValidation](#) ()  
*Finish the validation process. This is device specific and must be overwritten by the device class.*
- abstract Task< List< [GazeCalibrationData](#) > > [ApplyCalibration](#) ()  
*Apply the calibration data. This is device specific and must be overwritten by the device class.*
- abstract ? [GazeValidationData](#) [ComputeValidation](#) ()  
*Apply the validation data. This is device specific and must be overwritten by the device class.*
- abstract Task< bool > [CollectCalibrationDataAsync](#) (Point point)  
*Collect calibration data on a calibration point. This is device specific and must be overwritten by the device class.*
- abstract Task< bool > [CollectValidationDataAsync](#) (Point point)  
*Collect validation data on a validation point. This is device specific and must be overwritten by the device class.*
- void [StartDriftCompensation](#) ()  
*Start the drift compensation process.*
- void [ResetDriftCompensation](#) ()  
*Reset the drift compensation value*
- virtual bool [IsInitialised](#) ()  
*Checks wheter the device is connected and initialised. This is device specific and may be overwritten. Otherwise true is always returned.*

## Public Attributes

- readonly string [DeviceName](#)  
*The name of the tracker device*

## Protected Member Functions

- abstract void [InitDriftCompensation](#) ()  
*Initialise the drift compensation. This is device specific and must be overwritten by the device class.*
- abstract int [GetFixationFrameCount](#) (int durationThreshold)  
*Get the number of required gaze samples to compute a fixation. This is device specific and must be overwritten by the device because the duration of fixation point detection depends on the frame rate of the device.*
- abstract Vector3 [GetUnitDirection](#) ()  
*Get the unit vector pointing in the direction of the gaze vector. This is device specific as the gaze data are represented in a coordinate system as defined by the device.*
- virtual void [Dispose](#) (bool disposing)  
*Releases unmanaged and - optionally - managed resources.*
- bool [IsReady](#) ()  
*Determines whether this eye tracker is ready.*
- virtual void [OnGazeDataReceived](#) ([GazeData](#) gazeData)  
*Called when [gaze data received].*
- virtual void [OnUserPositionDataReceived](#) ([UserPositionData](#) e)  
*Called when [user position data received].*
- virtual void [OnPropertyChanged](#) (string property\_name)  
*Called when when the state property of EyeTracker is changing.*
- virtual void [OnTrackerDisabled](#) (EventArgs e)  
*Raises the E:TrackerDisabled event.*
- void [OnTrackerDisabledTimeout](#) (object? source, ElapsedEventArgs e)  
*Called after a specified amount of time of the eyetracker not being ready.*
- virtual void [OnTrackerEnabled](#) (EventArgs e)  
*Raises the E:TrackerEnabled event.*

## Protected Attributes

- Timer? [dialogBoxTimer](#)  
*Timer to control the apperance of the dialog box*
- [TrackerLogger](#) logger  
*The logger*
- [TrackerMessageBox?](#) [trackerMessageBox](#)  
*The dialog box that is controlled by the dialogBoxTimer*
- [DriftCompensation?](#) [driftCompensation](#)  
*drift compensation handler*
- [ScreenArea?](#) [screenArea](#) = null  
*The screen area structure holding the metrics of the screen in 3d space.*
- [ConfigItem](#) config  
*The gaze configuration item*

## Properties

- [ScreenArea?](#) [ScreenArea](#) [get]  
*The screen area structure holding the metrics of the screen in 3d space.*
- [DeviceStatus](#) [State](#) [get, set]  
*Gets or sets the state of the eyetracker device.*

## Events

- EventHandler? [TrackerEnabled](#)  
*Occurs when [tracker enabled].*
- EventHandler? [TrackerDisabled](#)  
*Occurs when [tracker disabled].*
- PropertyChangedEventHandler? [PropertyChanged](#)  
*Occurs when a property value changes.*
- [GazeDataHandler](#)? [GazeDataReceived](#)  
*Occurs when [gaze data received].*
- [DriftCompensationEventHandler](#)? [DriftCompensationComputed](#)  
*Occurs when drift compensation was computed.*
- [UserPositionDataHandler](#)? [UserPositionDataReceived](#)  
*Occurs when [user position data received].*

### 8.5.1 Detailed Description

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

See also

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

### 8.5.2 Member Enumeration Documentation

#### 8.5.2.1 DeviceStatus

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

The tracker device status

### 8.5.3 Constructor & Destructor Documentation

#### 8.5.3.1 BaseTracker()

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

Initializes a new instance of the EyeTrackerHandler class.

**Parameters**

<i>logger</i>	The logger.
<i>config</i>	The configuration object.
<i>deviceName</i>	Name of the device.

## 8.5.4 Member Function Documentation

### 8.5.4.1 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](#).

### 8.5.4.2 CollectCalibrationDataAsync()

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

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

**Parameters**

<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](#).

#### 8.5.4.3 CollectValidationDataAsync()

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

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

**Parameters**

<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](#).

**8.5.4.4 ComputeValidation()**

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

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

**Returns**

The validation data result.

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

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

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

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

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

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

Releases unmanaged and - optionally - managed resources.

**Parameters**

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

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



#### 8.5.4.7 DriftCompensationEventHandler()

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

Event handler for drift compensation events

##### Parameters

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

#### 8.5.4.8 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](#).

#### 8.5.4.9 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](#).

#### 8.5.4.10 FinishValidation()

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

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

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

#### 8.5.4.11 GazeDataHandler()

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

Event handler for gaze data events of the eyetracker

## Parameters

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

**8.5.4.12 GetFixationFrameCount()**

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

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

## Parameters

<i>durationThreshold</i>	The required fixation duration in milliseconds.
--------------------------	---

## Returns

The number of gaze samples to require for fixation detection.

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

**8.5.4.13 GetUnitDirection()**

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

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

## Returns

The unit vector

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

**8.5.4.14 InitCalibration()**

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

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

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

#### 8.5.4.15 InitCalibrationAsync()

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

Initialise the async 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](#).

#### 8.5.4.16 InitDriftCompensation()

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

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

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

#### 8.5.4.17 InitValidation()

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

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

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

#### 8.5.4.18 IsInitialised()

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

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

##### Returns

True

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

#### 8.5.4.19 IsReady()

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

Determines whether this eye tracker is ready.

##### Returns

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

#### 8.5.4.20 OnGazeDataReceived()

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

Called when [gaze data received].

##### Parameters

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

#### 8.5.4.21 OnPropertyChanged()

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

Called when when the state property of EyeTracker is changing.

##### Parameters

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

#### 8.5.4.22 OnTrackerDisabled()

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

Raises the E:TrackerDisabled event.

##### Parameters

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

#### 8.5.4.23 OnTrackerDisabledTimeout()

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

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

##### Parameters

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

#### 8.5.4.24 OnTrackerEnabled()

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

Raises the E:TrackerEnabled event.

##### Parameters

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

#### 8.5.4.25 OnUserPositionDataReceived()

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

Called when [user position data received].

##### Parameters

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

#### 8.5.4.26 PatternReplace()

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

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

#### Returns

The string where patterns were replaced.

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

#### 8.5.4.27 ResetDriftCompensation()

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

Reset the drift compensation value

#### 8.5.4.28 StartDriftCompensation()

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

Start the drift compensation process.

#### 8.5.4.29 UserPositionDataHandler()

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

Event handler for user position data events of the eyetracker

#### Parameters

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

### 8.5.5 Member Data Documentation

#### 8.5.5.1 config

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

The gaze configuration item

#### 8.5.5.2 DeviceName

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

The name of the tracker device

#### 8.5.5.3 dialogBoxTimer

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

Timer to control the apperance of the dialog box

#### 8.5.5.4 driftCompensation

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

drift compensation handler

#### 8.5.5.5 logger

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

The logger

#### 8.5.5.6 screenArea

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

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

#### 8.5.5.7 trackerMessageBox

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

The dialog box that is controlled by the dialogBoxTimer

## 8.5.6 Property Documentation

### 8.5.6.1 ScreenArea

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

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

### 8.5.6.2 State

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

Gets or sets the state of the eyetracker device.

The state.

## 8.5.7 Event Documentation

### 8.5.7.1 DriftCompensationComputed

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

Occurs when drift compensation was computed.

### 8.5.7.2 GazeDataReceived

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

Occurs when [gaze data received].

### 8.5.7.3 PropertyChanged

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

Occurs when a property value changes.



#### 8.5.7.4 TrackerDisabled

EventHandler? GazeUtilityLibrary.Tracker.BaseTracker.TrackerDisabled

Occurs when [tracker disabled].

#### 8.5.7.5 TrackerEnabled

EventHandler? GazeUtilityLibrary.Tracker.BaseTracker.TrackerEnabled

Occurs when [tracker enabled].

#### 8.5.7.6 UserPositionDataReceived

UserPositionDataHandler? GazeUtilityLibrary.Tracker.BaseTracker.UserPositionDataReceived

Occurs when [user position data received].

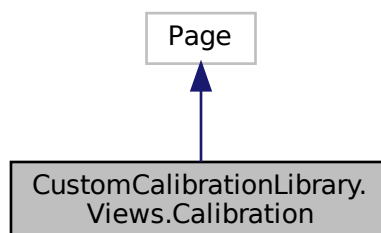
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Tracker/BaseTracker.cs

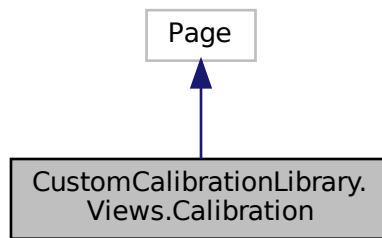
## 8.6 CustomCalibrationLibrary.Views.Calibration Class Reference

Interaction logic for Calibration.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.Calibration:



Collaboration diagram for CustomCalibrationLibrary.Views.Calibration:



## Public Member Functions

- [Calibration](#) ([CalibrationModel](#) model)  
*Initializes a new instance of the [Calibration](#) class.*

### 8.6.1 Detailed Description

Interaction logic for Calibration.xaml

### 8.6.2 Constructor & Destructor Documentation

#### 8.6.2.1 Calibration()

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

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

#### Parameters

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

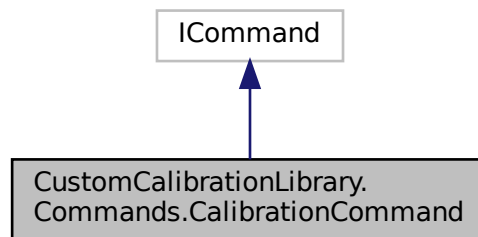
The documentation for this class was generated from the following file:

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

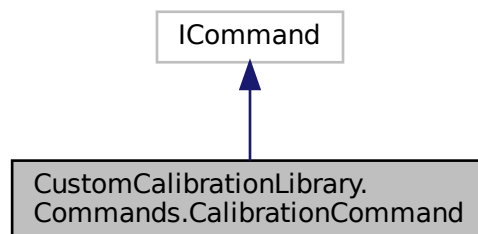
## 8.7 CustomCalibrationLibrary.Commands.CalibrationCommand Class Reference

Command class to trigger calibration events.

Inheritance diagram for CustomCalibrationLibrary.Commands.CalibrationCommand:



Collaboration diagram for CustomCalibrationLibrary.Commands.CalibrationCommand:



### Public Member Functions

- [CalibrationCommand](#) ([CalibrationModel](#) model, [CalibrationEventType](#) eventType)  
*Initializes a new instance of the [CalibrationCommand](#) class.*
- bool [CanExecute](#) (object? parameter)  
*Returns whether command can be executed or not.*
- void [Execute](#) (object? parameter)  
*Send calibration event.*

### Properties

- EventHandler? [CanExecuteChanged](#)  
*Event handler on can executed flag change.*

### 8.7.1 Detailed Description

Comand class to trigger calibration events.

### 8.7.2 Constructor & Destructor Documentation

#### 8.7.2.1 CalibrationCommand()

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

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

##### Parameters

<i>model</i>	The calibration model
<i>eventType</i>	The type of the calibration event.

### 8.7.3 Member Function Documentation

#### 8.7.3.1 CanExecute()

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

Returns whether command can be executed or not.

##### Parameters

<i>parameter</i>	The command parameter
------------------	-----------------------

##### Returns

True

#### 8.7.3.2 Execute()

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

Send calibration event.

## Parameters

<i>parameter</i>	The command parameter
------------------	-----------------------

## 8.7.4 Property Documentation

### 8.7.4.1 CanExecuteChanged

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

Event handler on can executed flag change.

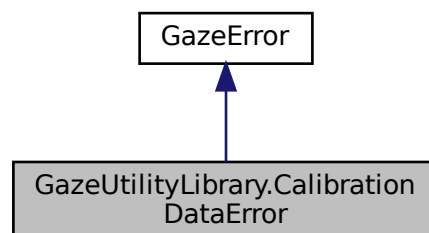
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Commands/CalibrationCommand.cs

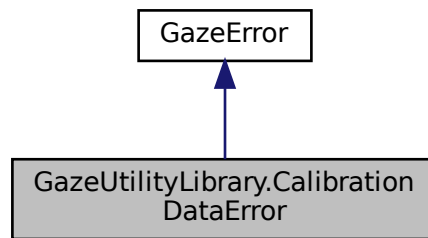
## 8.8 GazeUtilityLibrary.CalibrationDataError Class Reference

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

Inheritance diagram for GazeUtilityLibrary.CalibrationDataError:



Collaboration diagram for GazeUtilityLibrary.CalibrationDataError:



## Public Member Functions

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

## Properties

- [ECalibrationDataError](#) Error [set]  
*The error flags.*

## Additional Inherited Members

### 8.8.1 Detailed Description

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

### 8.8.2 Member Function Documentation

#### 8.8.2.1 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

### 8.8.3 Property Documentation

#### 8.8.3.1 Error

`ECalibrationDataError` GazeUtilityLibrary.CalibrationDataError.Error [set]

The error flags.

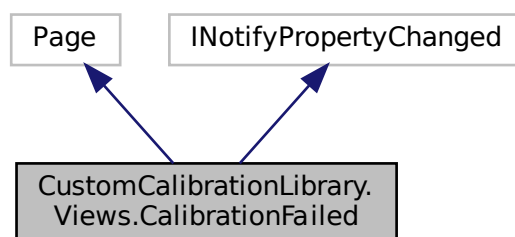
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeError.cs

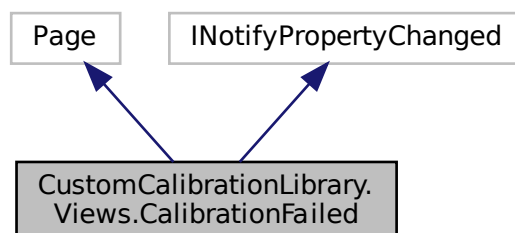
## 8.9 CustomCalibrationLibrary.Views.CalibrationFailed Class Reference

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.*

### 8.9.1 Detailed Description

Interaction logic for CalibrationFailed.xaml

### 8.9.2 Constructor & Destructor Documentation

#### 8.9.2.1 CalibrationFailed()

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

Constructor

Parameters

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

### 8.9.3 Property Documentation



### 8.9.3.1 CalibrationAbortCommand

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

Command to abort the calibration

### 8.9.3.2 CalibrationRestartCommand

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

Command to restart the calibration

### 8.9.3.3 Error

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

The error message to be updated on the view.

## 8.9.4 Event Documentation

### 8.9.4.1 PropertyChanged

`PropertyChangedEventHandler? CustomCalibrationLibrary.Views.CalibrationFailed.PropertyChanged`

The property change event to update the view.

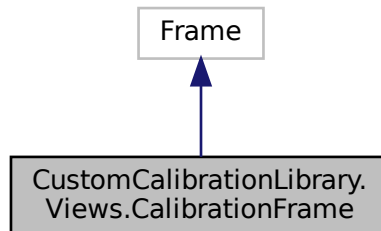
The documentation for this class was generated from the following file:

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

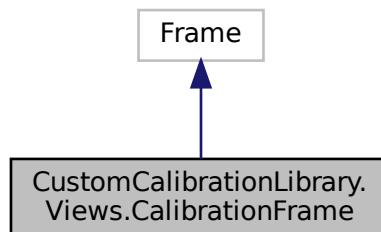
## 8.10 CustomCalibrationLibrary.Views.CalibrationFrame Class Reference

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

#### 8.10.1 Detailed Description

Interaction logic for CalibrationCollection.xaml

#### 8.10.2 Constructor & Destructor Documentation

### 8.10.2.1 CalibrationFrame()

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

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

#### Parameters

<i>model</i>	The calibration model.
<i>window</i>	The target window.

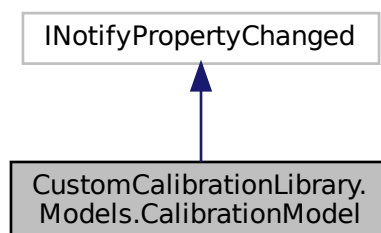
The documentation for this class was generated from the following file:

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

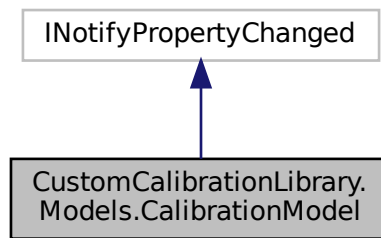
## 8.11 CustomCalibrationLibrary.Models.CalibrationModel Class Reference

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)  
*The calibraion event change handler.*
- `CalibrationModel` (`TrackerLogger` logger, double[ ][ ] points)  
*Initializes a new instance of the `CalibrationModel` class.*
- 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]  
*The data returned by a successful validation process.*
- `CalibrationStatus Status` [get, set]  
*The status of the calibarion process.*
- `CalibrationStatus LastStatus` [get]  
*The calibration status before an error occured.*
- Point[ ] `Points` [get]  
*All calibration points.*
- ObservableCollection< `CalibrationPoint` > `CalibrationPoints` [get]  
*The calibration points to be added during the calibration process.*

- Point [GazePoint](#) [get]  
*The gaze point position.*
- [UserPositionData](#) [UserPositionGuide](#) [get, set]  
*The user position giude values.*
- int [Index](#) [get]  
*The index of the current calibration point*

## Events

- EventHandler< [CalibrationEventType](#) >? [CalibrationEvent](#)  
*Event to trigger changes in the calibration process.*
- PropertyChangedEventHandler? [PropertyChanged](#)  
*Event to trigger property changes in this class.*
- EventHandler< Point >? [GazePointChanged](#)  
*Event to trigger gaze point changes.*
- EventHandler< [UserPositionData](#) >? [UserPositionGuideChanged](#)  
*Event to trigger user position guide changes.*

### 8.11.1 Detailed Description

The model for the calibration process.

### 8.11.2 Constructor & Destructor Documentation

#### 8.11.2.1 CalibrationModel()

```
CustomCalibrationLibrary.Models.CalibrationModel.CalibrationModel (
    TrackerLogger logger,
    double points[ ][ ] ) [inline]
```

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

#### Parameters

<i>logger</i>	The log handler.
<i>points</i>	Calibration points.

### 8.11.3 Member Function Documentation

#### 8.11.3.1 GazeDataCollected()

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

Trigger the data collected events.

#### 8.11.3.2 InitCalibration()

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

Initialise the calibration.

#### 8.11.3.3 NextCalibrationPoint()

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

Trigger the next calibration point.

#### 8.11.3.4 OnCalibrationEvent()

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

The calibraion event change handler.

##### Parameters

<i>type</i>	
-------------	--

#### 8.11.3.5 RedoCalibrationPoint()

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

Remove and re-add the current calibration point

#### 8.11.3.6 SetCalibrationResult()

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

Updates the calibration results on the screen.

#### Parameters

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

### 8.11.3.7 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

## 8.11.4 Property Documentation

### 8.11.4.1 CalibrationPoints

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

The calibration points to be added during the calibration process.

### 8.11.4.2 Error

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

The error message of the calibration process.

### 8.11.4.3 GazePoint

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

The gaze point position.

#### 8.11.4.4 Index

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

The index of the current calibration point

#### 8.11.4.5 LastStatus

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

The calibration status before an error occurred.

#### 8.11.4.6 Points

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

All calibration points.

#### 8.11.4.7 Status

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

The status of the calibration process.

#### 8.11.4.8 UserPositionGuide

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

The user position guide values.

#### 8.11.4.9 ValidationData

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

The data returned by a successful validation process.



## 8.11.5 Event Documentation

### 8.11.5.1 CalibrationEvent

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

Event to trigger changes in the calibration process.

### 8.11.5.2 GazePointChanged

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

Event to trigger gaze point changes.

### 8.11.5.3 PropertyChanged

```
PropertyChangedEventHandler? CustomCalibrationLibrary.Models.CalibrationModel.PropertyChanged
```

Event to trigger property changes in this class.

### 8.11.5.4 UserPositionGuideChanged

```
EventHandler<UserPositionData>? CustomCalibrationLibrary.Models.CalibrationModel.UserPosition↔  
GuideChanged
```

Event to trigger user position guide changes.

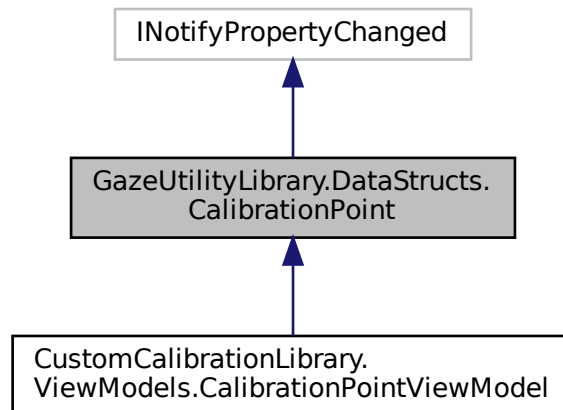
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Models/CalibrationModel.cs

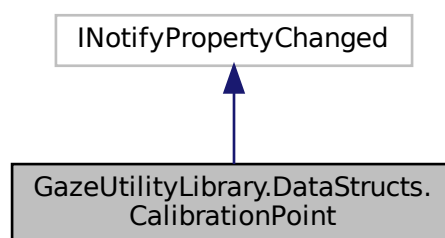
## 8.12 GazeUtilityLibrary.DataStructs.CalibrationPoint Class Reference

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)

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

## Properties

- int [Index](#) [get]  
*The index of the calibration point.*
- bool [HasData](#) [get, set]  
*Flag to indicate whether data has been collected for this calibration point.*
- 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](#)  
*Event to trigger property changes.*

### 8.12.1 Detailed Description

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

### 8.12.2 Constructor & Destructor Documentation

#### 8.12.2.1 CalibrationPoint()

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

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

#### Parameters

<i>position</i>	The position of the calibration point.
<i>index</i>	The index of the calibration point.

### 8.12.3 Property Documentation

### 8.12.3.1 GazePositionAverage

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

The average between the left and the right gaze point.

### 8.12.3.2 GazePositionLeft

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

The left gaze point.

### 8.12.3.3 GazePositionRight

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

The right gaze point.

### 8.12.3.4 HasData

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

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

### 8.12.3.5 Index

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

The index of the calibration point.

### 8.12.3.6 Position

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

The position of the calibration point.

## 8.12.4 Event Documentation

### 8.12.4.1 PropertyChanged

`PropertyChangedEventHandler? GazeUtilityLibrary.DataStructs.CalibrationPoint.PropertyChanged`

Event to trigger property changes.

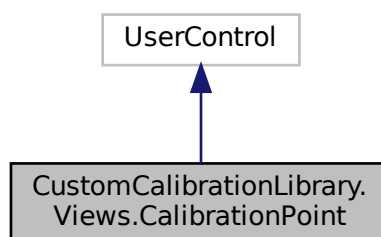
The documentation for this class was generated from the following file:

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

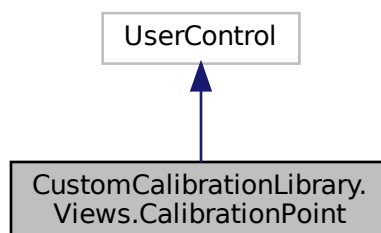
## 8.13 CustomCalibrationLibrary.Views.CalibrationPoint Class Reference

Interaction logic for CalibrationPoint.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationPoint:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationPoint:



## Public Member Functions

- [CalibrationPoint](#) ()  
*Initializes a new instance of the [CalibrationPoint](#) class.*

### 8.13.1 Detailed Description

Interaction logic for CalibrationPoint.xaml

### 8.13.2 Constructor & Destructor Documentation

#### 8.13.2.1 CalibrationPoint()

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

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

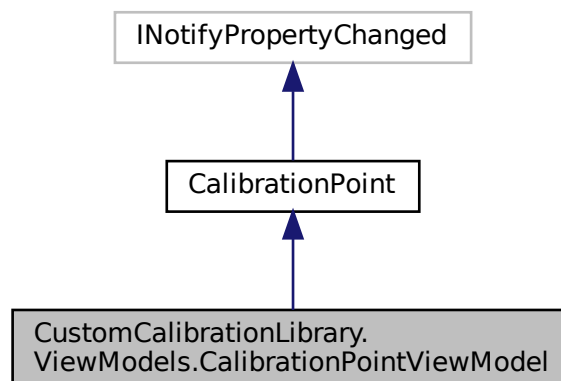
The documentation for this class was generated from the following file:

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

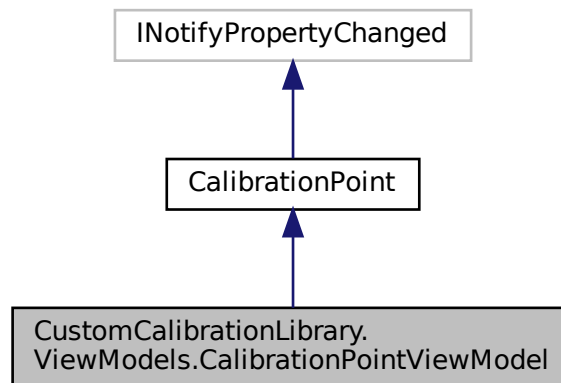
## 8.14 CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel Class Reference

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)  
*Initializes a new instance of the [CalibrationPointViewModel](#) class.*
- [CalibrationPointViewModel](#) (CalibrationPoint point)  
*Initializes a new instance of the [CalibrationPointViewModel](#) class.*

## Additional Inherited Members

### 8.14.1 Detailed Description

The view model for a calibration point.

### 8.14.2 Constructor & Destructor Documentation

#### 8.14.2.1 CalibrationPointViewModel() [1/2]

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

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

## Parameters

<i>point</i>	The position of the calibration point.
<i>index</i>	The index of the calibration point.

**8.14.2.2 CalibrationPointViewModel()** [2/2]

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

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

## Parameters

<i>point</i>	The calibration point object.
--------------	-------------------------------

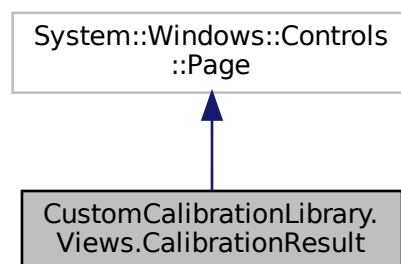
The documentation for this class was generated from the following file:

- [source/CustomCalibrationLibrary/ViewModels/CalibrationPointViewModel.cs](#)

**8.15 CustomCalibrationLibrary.Views.CalibrationResult Class Reference**

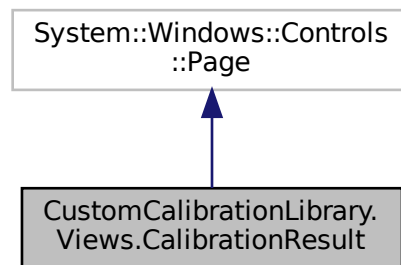
Interaction logic for CalibrationResult.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationResult:





Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationResult:



## Public Member Functions

- [CalibrationResult](#) ([CalibrationModel](#) model)  
*Initializes a new instance of the [CalibrationResult](#) class.*

### 8.15.1 Detailed Description

Interaction logic for CalibrationResult.xaml

### 8.15.2 Constructor & Destructor Documentation

#### 8.15.2.1 CalibrationResult()

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

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

#### Parameters

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

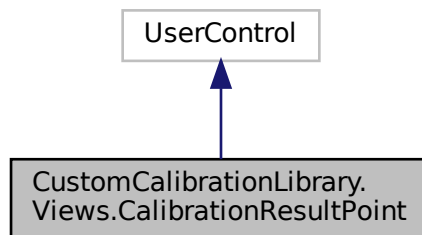
The documentation for this class was generated from the following file:

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

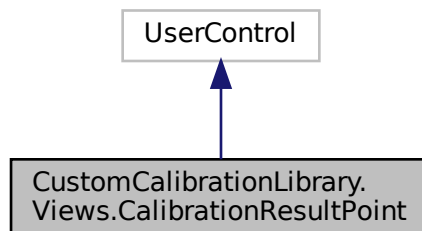
## 8.16 CustomCalibrationLibrary.Views.CalibrationResultPoint Class Reference

Interaction logic for CalibrationResultPoint.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationResultPoint:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationResultPoint:



### Public Member Functions

- [CalibrationResultPoint](#) ()  
*Initializes a new instance of the [CalibrationResultPoint](#) class.*

### 8.16.1 Detailed Description

Interaction logic for CalibrationResultPoint.xaml

### 8.16.2 Constructor & Destructor Documentation

### 8.16.2.1 CalibrationResultPoint()

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

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

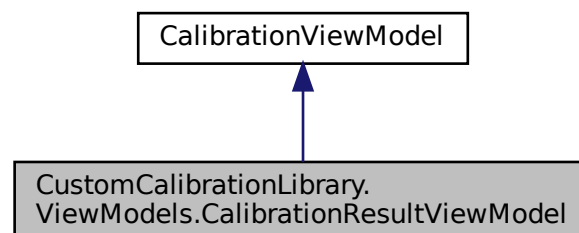
The documentation for this class was generated from the following file:

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

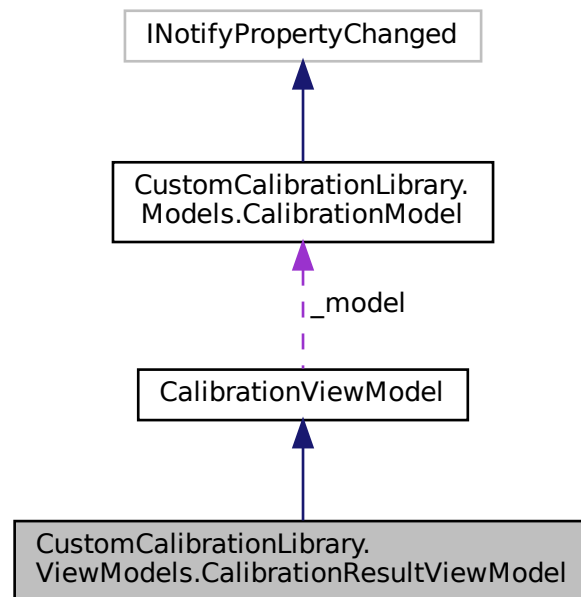
## 8.17 CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel Class Reference

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

### 8.17.1 Detailed Description

View model class of the gaze calibration result.

## 8.17.2 Constructor & Destructor Documentation

### 8.17.2.1 CalibrationResultViewModel()

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

Constructor

Parameters

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

## 8.17.3 Member Function Documentation

### 8.17.3.1 OnGazeToggle()

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

Toggle the visibility of the live gaze point.

## 8.17.4 Property Documentation

### 8.17.4.1 CalibrationAcceptCommand

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

Command to accept the calibration

### 8.17.4.2 CalibrationRestartCommand

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

Command to restart the calibration

### 8.17.4.3 GazePoint

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

The position of the live gaze point

### 8.17.4.4 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

## 8.18 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` ()  
*Convert validation values to a string.*

### Properties

- NormalizedPoint2D `Coordinates` [get]  
*The 2D coordinates of this point (in Active Display Coordinate System).*
- float `AccuracyLeftEye` [get]  
*The accuracy in degrees for the left eye.*
- float `PrecisionLeftEye` [get]  
*The precision (standard deviation) in degrees for the left eye.*
- float `PrecisionRMSLeftEye` [get]  
*The precision (root mean square of sample-to-sample error) in degrees for the left eye.*
- float `AccuracyRightEye` [get]  
*The accuracy in degrees for the right eye.*
- float `PrecisionRightEye` [get]  
*The precision (standard deviation) in degrees for the right eye.*
- float `PrecisionRMSRightEye` [get]  
*The precision (root mean square of sample-to-sample error) in degrees for the right eye.*
- bool `TimedOut` [get]  
*A boolean indicating if there was a timeout while collecting data for this point.*
- GazeDataEventArgs[] `GazeData` [get]  
*The gaze data samples collected for this point. These samples are the base for the calculated accuracy and precision.*

### 8.18.1 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.

### 8.18.2 Member Function Documentation

#### 8.18.2.1 ToString()

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

Convert validation values to a string.

##### Returns

The validation string.

### 8.18.3 Property Documentation

#### 8.18.3.1 AccuracyLeftEye

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

The accuracy in degrees for the left eye.

#### 8.18.3.2 AccuracyRightEye

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

The accuracy in degrees for the right eye.

#### 8.18.3.3 Coordinates

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

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

#### 8.18.3.4 GazeData

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

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

#### 8.18.3.5 PrecisionLeftEye

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

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

#### 8.18.3.6 PrecisionRightEye

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

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

#### 8.18.3.7 PrecisionRMSLeftEye

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

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

#### 8.18.3.8 PrecisionRMSRightEye

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

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

#### 8.18.3.9 TimedOut

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

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

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

- `source/TobiiProSdkAddons/ScreenBasedCalibrationValidation.cs`



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

Contains the result of the calibration validation.

### Public Member Functions

- override string [ToString](#) ()  
*Convert validation values to a string.*

### Properties

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

### 8.19.1 Detailed Description

Contains the result of the calibration validation.

### 8.19.2 Member Function Documentation

#### 8.19.2.1 ToString()

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

Convert validation values to a string.

#### Returns

The validation string.

### 8.19.3 Property Documentation

#### 8.19.3.1 AverageAccuracyLeftEye

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

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

#### 8.19.3.2 AverageAccuracyRightEye

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

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

#### 8.19.3.3 AveragePrecisionLeftEye

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

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

#### 8.19.3.4 AveragePrecisionRightEye

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

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

#### 8.19.3.5 AveragePrecisionRMSLeftEye

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

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

### 8.19.3.6 AveragePrecisionRMSRightEye

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

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

### 8.19.3.7 Points

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

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

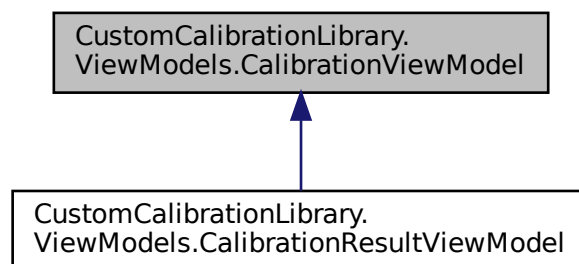
The documentation for this class was generated from the following file:

- source/TobiiProSdkAddons/ScreenBasedCalibrationValidation.cs

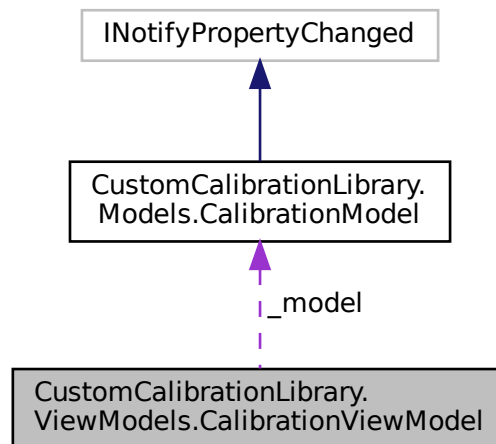
## 8.20 CustomCalibrationLibrary.ViewModels.CalibrationViewModel Class Reference

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`  
*The claibration model.*

## Properties

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

### 8.20.1 Detailed Description

The view model class of the calibration view

### 8.20.2 Constructor & Destructor Documentation

#### 8.20.2.1 CalibrationViewModel()

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

Constructor

## Parameters

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

## 8.20.3 Member Data Documentation

### 8.20.3.1 `_model`

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

The claibration model.

## 8.20.4 Property Documentation

### 8.20.4.1 CalibrationPoints

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

The collection of calibration points to be shown on the view

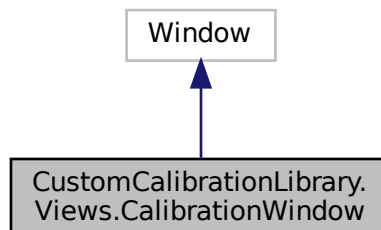
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/CalibrationViewModel.cs

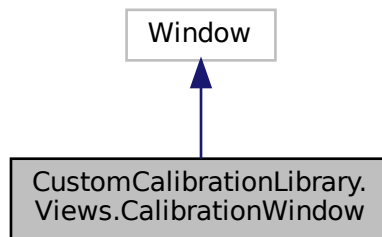
## 8.21 CustomCalibrationLibrary.Views.CalibrationWindow Class Reference

Interaction logic for MainWindow.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.CalibrationWindow:



Collaboration diagram for CustomCalibrationLibrary.Views.CalibrationWindow:



### 8.21.1 Detailed Description

Interaction logic for MainWindow.xaml

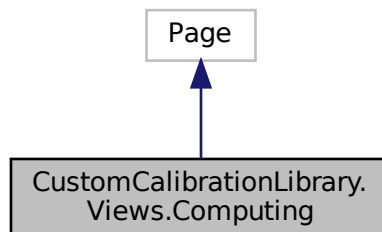
The documentation for this class was generated from the following file:

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

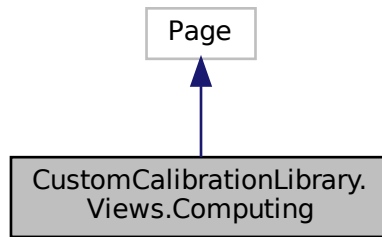
## 8.22 CustomCalibrationLibrary.Views.Computing Class Reference

Interaction logic for Computing.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.Computing:



Collaboration diagram for CustomCalibrationLibrary.Views.Computing:



## Public Member Functions

- [Computing](#) ()  
*Initializes a new instance of the [Computing](#) class.*

### 8.22.1 Detailed Description

Interaction logic for Computing.xaml

### 8.22.2 Constructor & Destructor Documentation

#### 8.22.2.1 Computing()

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

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

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

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

## 8.23 GazeUtilityLibrary.ConfigItem Class Reference

configuration file class

## Public Member Functions

- [ConfigItem](#) ()  
*Initializes a new instance of the [ConfigItem](#) class.*

## Properties

- string [CalibrationLogColumnOrder](#) [get, set]  
*Allows to define the order and the delimiters between the different calibration data values.*
- string[] [CalibrationLogColumnTitle](#) [get, set]  
*Defines the titles of the calibration data log value columns.*
- bool [CalibrationLogWriteOutput](#) [get, set]  
*Defines whether gaze calibration data is written to a log file.*
- double[][] [CalibrationPoints](#) [get, set]  
*Define the calibration points to be shown during the calibration process.*
- double [DriftCompensationDispersionThreshold](#) [get, set]  
*In order to detect a fixation with the I-DT algorithm a dispersion threshold is required. Provide an angle in degrees.*
- double [DriftCompensationDispersionThresholdMax](#) [get, set]  
*In order to prevent drift compensation from getting out of hand limit the maximal allowed dispresion. If the drift compensation angle is larger than the here defined degrees, no compensation is applied. Provide an angle in degrees.*
- int [DriftCompensationDurationThreshold](#) [get, set]  
*Specifies the amount of time (in milliseconds) required to fixate the target during drift compensation.*
- int [DriftCompensationTimer](#) [get, set]  
*Specifies the amount of time (in milliseconds) to wait for a fixation point during drift compensation.*
- bool [DriftCompensationWindowShow](#) [get, set]  
*If set to true the drift compensation window is shown on the drift compensation command. Otherwise only the drift compensation process is done without showing the window.*
- int [ValidationDurationThreshold](#) [get, set]  
*Specifies the amount of time (in milliseconds) required to fixate the target during validation.*
- string [ValidationLogColumnOrder](#) [get, set]  
*Allows to define the order and the delimiters between the different validation data values.*
- string[] [ValidationLogColumnTitle](#) [get, set]  
*Defines the titles of the validation data log value columns.*
- bool [ValidationLogWriteOutput](#) [get, set]  
*Defines whether gaze validation data is written to a log file.*
- double[][] [ValidationPoints](#) [get, set]  
*Define the validation points to be shown during the validation process.*
- int [ValidationTimer](#) [get, set]  
*Specifies the amount of time (in milliseconds) to wait for a fixation point during validation.*
- string [DataLogColumnOrder](#) [get, set]  
*Allows to define the order and the delimiters between the different gaze data values.*
- string[] [DataLogColumnTitle](#) [get, set]  
*Defines the titles of the gaze data log value columns.*
- int [DataLogCount](#) [get, set]  
*Number of maximal allowed output data files in the output path. Oldest files are deleted first.*
- string [DataLogFormatDiameter](#) [get, set]  
*Allows to define the format of how the pupil diameter (in millimetres) will be logged.*
- string [DataLogFormatOrigin](#) [get, set]  
*Allows to define the format of how the gaze origin values (in millimetres) will be logged.*
- string [DataLogFormatNormalizedPoint](#) [get, set]  
*Allows to define the format of how normalized data points will be logged.*



- string [DataLogFormatTimeStamp](#) [get, set]  
*Allows to define the format of the timestamp.*
- string [DataLogFormatTimeStampRelative](#) [get, set]  
*Allows to define the format of the relative timestamp in milliseconds.*
- string [DataLogFormatValidation](#) [get, set]  
*Allows to define the format of the validation values.*
- string [DataLogPath](#) [get, set]  
*Defines the location of the output file. It must be the path to a folder (not a file).*
- bool [DataLogWriteOutput](#) [get, set]  
*Defines whether gaze data is written to a log file.*
- bool [DataLogDisabledOnStartup](#) [get, set]  
*Defines whether gaze data storing is disabled on Gaze application start.*
- string? [ConfigName](#) [get, set]  
*The name of the experiment.*
- string? [LicensePath](#) [get, set]  
*Defines the location of the license files. It must be the path to a folder (not a file).*
- int [ReadyTimer](#) [get, set]  
*Specifies the amount of time (in milliseconds) to wait for the eye tracker to become ready while it is in any other state.*
- int [TrackerDevice](#) [get, set]  
*Choose the tracker device (1: [Tobii Pro SDK](#), 2: Mouse [Tracker](#)).*
- bool [MouseControl](#) [get, set]  
*Defines whether the mouse cursor shall be controlled by the gaze of the subject during the experiment.*
- bool [MouseControlHide](#) [get, set]  
*Defines whether the mouse cursor shall be hidden during the experiment.*
- bool [MouseCalibrationHide](#) [get, set]  
*Defines whether the mouse cursor shall be hidden on the calibration window.*
- string [MouseStandardIconPath](#) [get, set]  
*Defines the Path to the standard mouse pointer icon.*
- string [TobiiApplicationPath](#) [get, set]  
*Defines the [Tobii](#) installation path. It must be the path to a folder (not a file).*
- string [TobiiCalibrate](#) [get, set]  
*The [Tobii](#) application to run a calibration.*
- string [TobiiCalibrateArguments](#) [get, set]  
*The arguments to pass to the calibration application. Use S as a placeholder for the device serial number and A as a placeholder for the device address.*
- [ConfigScreenArea](#) [ScreenArea](#) [get, set]  
*Hold the screen area once the config file is dumped during experimentation.*

### 8.23.1 Detailed Description

configuration file class

### 8.23.2 Constructor & Destructor Documentation

### 8.23.2.1 ConfigItem()

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

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

## 8.23.3 Property Documentation

### 8.23.3.1 CalibrationLogColumnOrder

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

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

### 8.23.3.2 CalibrationLogColumnTitle

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

Defines the titles of the calibration data log value columns.

### 8.23.3.3 CalibrationLogWriteOutput

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

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

### 8.23.3.4 CalibrationPoints

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

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

### 8.23.3.5 ConfigName

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

The name of the experiment.

### 8.23.3.6 DataLogColumnOrder

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

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

### 8.23.3.7 DataLogColumnTitle

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

Defines the titles of the gaze data log value columns.

### 8.23.3.8 DataLogCount

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

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

### 8.23.3.9 DataLogDisabledOnStartup

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

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

### 8.23.3.10 DataLogFormatDiameter

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

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

### 8.23.3.11 DataLogFormatNormalizedPoint

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

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

#### 8.23.3.12 DataLogFormatOrigin

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

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

#### 8.23.3.13 DataLogFormatTimeStamp

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

Allows to define the format of the timestamp.

#### 8.23.3.14 DataLogFormatTimeStampRelative

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

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

#### 8.23.3.15 DataLogFormatValidation

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

Allows to define the format of the validation values.

#### 8.23.3.16 DataLogPath

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

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

#### 8.23.3.17 DataLogWriteOutput

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

Defines whether gaze data is written to a log file.

### 8.23.3.18 DriftCompensationDispersionThreshold

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

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

### 8.23.3.19 DriftCompensationDispersionThresholdMax

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

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

### 8.23.3.20 DriftCompensationDurationThreshold

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

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

### 8.23.3.21 DriftCompensationTimer

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

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

### 8.23.3.22 DriftCompensationWindowShow

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

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

### 8.23.3.23 LicensePath

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

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

#### 8.23.3.24 MouseCalibrationHide

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

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

#### 8.23.3.25 MouseControl

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

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

#### 8.23.3.26 MouseControlHide

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

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

#### 8.23.3.27 MouseStandardIconPath

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

Defines the Path to the standard mouse pointer icon.

#### 8.23.3.28 ReadyTimer

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

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

#### 8.23.3.29 ScreenArea

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

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

### 8.23.3.30 TobiiApplicationPath

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

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

### 8.23.3.31 TobiiCalibrate

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

The [Tobii](#) application to run a calibration.

### 8.23.3.32 TobiiCalibrateArguments

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

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

### 8.23.3.33 TrackerDevice

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

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

### 8.23.3.34 ValidationDurationThreshold

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

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

### 8.23.3.35 ValidationLogColumnOrder

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

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

### 8.23.3.36 ValidationLogColumnTitle

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

Defines the titles of the validation data log value columns.

### 8.23.3.37 ValidationLogWriteOutput

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

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

### 8.23.3.38 ValidationPoints

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

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

### 8.23.3.39 ValidationTimer

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

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

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

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

## 8.24 GazeUtilityLibrary.ConfigScreenArea Class Reference

The JSON structure of the screen area.

### Public Member Functions

- [ConfigScreenArea](#) ()  
*Initializes a new instance of the [ConfigScreenArea](#) class.*
- [ConfigScreenArea](#) ([ScreenArea](#) screenArea)  
*Initializes a new instance of the [ConfigScreenArea](#) class.*



## Properties

- double [Width](#) [get, set]  
*The width of the screen.*
- double [Height](#) [get, set]  
*The height of the screen.*
- double[] [Center](#) [get, set]  
*The coordinates of the center point of the screen.*
- double[] [TopLeft](#) [get, set]  
*The coordinates of the top left point of the screen.*
- double[] [TopRight](#) [get, set]  
*The coordinates of the to right point of the screen.*
- double[] [BottomLeft](#) [get, set]  
*The coordinates of the bottom left point of the screen.*
- double[] [BottomRight](#) [get, set]  
*The coordinates of the bottom right point of the screen.*

### 8.24.1 Detailed Description

The JSON structure of the screen area.

### 8.24.2 Constructor & Destructor Documentation

#### 8.24.2.1 ConfigScreenArea() [1/2]

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

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

#### 8.24.2.2 ConfigScreenArea() [2/2]

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

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

#### Parameters

<i>screenArea</i>	A screen area object.
-------------------	-----------------------

### 8.24.3 Property Documentation

#### 8.24.3.1 BottomLeft

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

The coordinates of the bottom left point of the screen.

#### 8.24.3.2 BottomRight

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

The coordinates of the bottom right point of the screen.

#### 8.24.3.3 Center

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

The coordinates of the center point of the screen.

#### 8.24.3.4 Height

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

The height of the screen.

#### 8.24.3.5 TopLeft

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

The coordinates of the top left point of the screen.

### 8.24.3.6 TopRight

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

The coordinates of the to right point of the screen.

### 8.24.3.7 Width

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

The width of the screen.

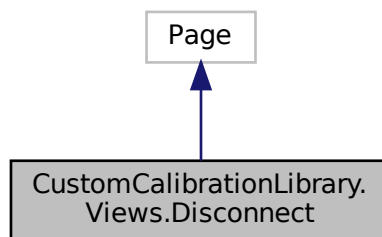
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeConfiguration.cs

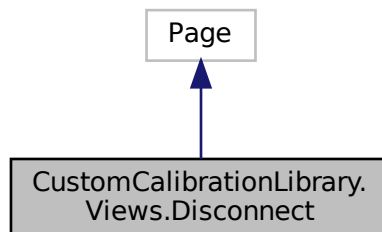
## 8.25 CustomCalibrationLibrary.Views.Disconnect Class Reference

Interaction logic for Disconnect.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.Disconnect:



Collaboration diagram for CustomCalibrationLibrary.Views.Disconnect:



## Public Member Functions

- [Disconnect](#) ([CalibrationModel](#) model)  
*Initializes a new instance of the [Disconnect](#) class.*

## Properties

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

### 8.25.1 Detailed Description

Interaction logic for Disconnect.xaml

### 8.25.2 Constructor & Destructor Documentation

#### 8.25.2.1 Disconnect()

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

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

##### Parameters

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

### 8.25.3 Property Documentation

#### 8.25.3.1 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

## 8.26 GazeUtilityLibrary.DriftCompensation Class Reference

The class to handle drift compensation.

### Public Member Functions

- [DriftCompensation](#) (Vector3 fixationPoint, int fixationFrameCount, double dispersionThreshold, double dispersionThresholdMax)  
*Initializes a new instance of the [DriftCompensation](#) class.*
- void [Reset](#) ()  
*Reset the drift compensation quaternion to the identity.*
- void [Start](#) ()  
*Start the drift compensation.*
- bool [Update](#) ([GazeData](#) gazeData)  
*Collect gaze data samples of a fixation and once enough samples are collected, compute the drift compensation quaternion.*

### Properties

- Quaternion [Q](#) [get]  
*The drift compensation quatrenion.*
- double [DeviationAngle](#) [get]  
*The deviation angle of the drift compensation.*
- double [Dispersion](#) [get]  
*The dispersion of the drift compensation fixation.*

#### 8.26.1 Detailed Description

The class to handle drift compensation.

#### 8.26.2 Constructor & Destructor Documentation

##### 8.26.2.1 DriftCompensation()

```
GazeUtilityLibrary.DriftCompensation.DriftCompensation (
    Vector3 fixationPoint,
    int fixationFrameCount,
    double dispersionThreshold,
    double dispersionThresholdMax ) [inline]
```

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

##### Parameters

<i>fixationPoint</i>	The target fixation point.
<i>fixationFrameCount</i>	The required number of frames during fixation.
<i>dispersionThreshold</i>	The dispersion threshold for the fixation.
<i>dispersionThresholdMax</i>	The maximal allowed deviation angle.

## 8.26.3 Member Function Documentation

### 8.26.3.1 Reset()

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

Reset the drift compensation quaternion to the identity.

### 8.26.3.2 Start()

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

Start the drift compensation.

### 8.26.3.3 Update()

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

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

#### Parameters

<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.

## 8.26.4 Property Documentation

### 8.26.4.1 DeviationAngle

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

The deviation angle of the drift compensation.

### 8.26.4.2 Dispersion

`double GazeUtilityLibrary.DriftCompensation.Dispersion [get]`

The dispersion of the drift compensation fixation.

### 8.26.4.3 Q

`Quaternion GazeUtilityLibrary.DriftCompensation.Q [get]`

The drift compensation quatrenion.

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

- `source/GazeUtilityLibrary/DriftCompensation.cs`

## 8.27 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*

### 8.27.1 Detailed Description

The drift compensation data structure

### 8.27.2 Constructor & Destructor Documentation

#### 8.27.2.1 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

### 8.27.3 Property Documentation

#### 8.27.3.1 Compensation

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

The drift compensation quaternion

#### 8.27.3.2 GazePosition2d

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

The drift compensated 2d gaze position

#### 8.27.3.3 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`

## 8.28 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.*

### 8.28.1 Detailed Description

The view model class of the drift compensation view.

### 8.28.2 Constructor & Destructor Documentation

#### 8.28.2.1 DriftCompensationViewModel()

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

Constructor

### 8.28.3 Property Documentation

#### 8.28.3.1 FixationPoint

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

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

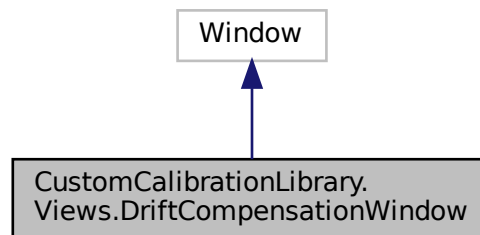
The documentation for this class was generated from the following file:

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

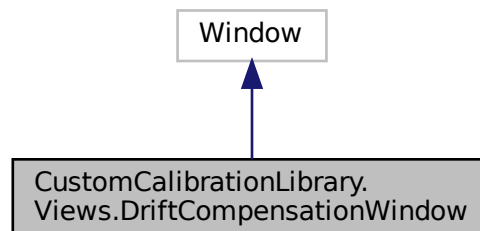
## 8.29 CustomCalibrationLibrary.Views.DriftCompensationWindow Class Reference

Interaction logic for DriftCompensation.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.DriftCompensationWindow:



Collaboration diagram for CustomCalibrationLibrary.Views.DriftCompensationWindow:



### Public Member Functions

- [DriftCompensationWindow](#) ()  
*Initializes a new instance of the [DriftCompensationWindow](#) class.*

### 8.29.1 Detailed Description

Interaction logic for DriftCompensation.xaml

### 8.29.2 Constructor & Destructor Documentation

### 8.29.2.1 DriftCompensationWindow()

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

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

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

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

## 8.30 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 the pupil diameter*

### 8.30.1 Detailed Description

The eye data set, including pupil information.

### 8.30.2 Constructor & Destructor Documentation

#### 8.30.2.1 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.

### 8.30.3 Property Documentation

#### 8.30.3.1 IsPupilDiameterValid

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

The validity flag of the pupil diameter

#### 8.30.3.2 PupilDiameter

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

The diameter of the pupil

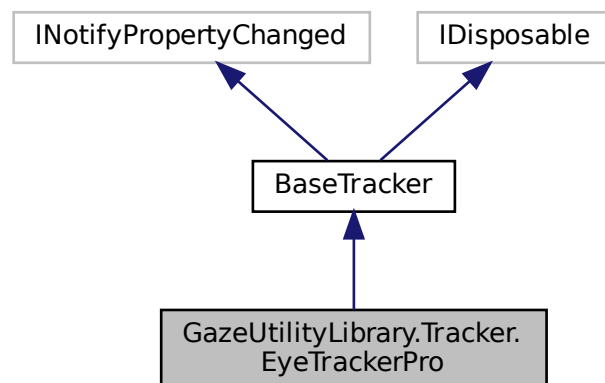
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/EyeData.cs

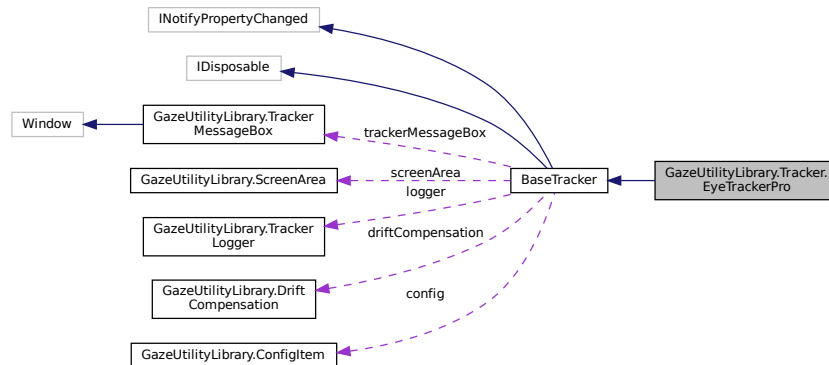
## 8.31 GazeUtilityLibrary.Tracker.EyeTrackerPro Class Reference

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 Tobi calibration result into the [GazeCalibrationData](#) structure.*
- override? [GazeValidationData](#) [ComputeValidation](#) ()  
*Compute the validation data.*
- bool [IsLicenseOk](#) ()  
*Determines whether the license is applied to the eyetracker device*
- override bool [IsInitialised](#) ()  
*Checks if the tracker device exists.*
- override string [PatternReplace](#) (string pattern)  
*Replaces a patten string with information from the eye tracker. Supported patterns are S for the serial number and A for the address.*

## Protected Member Functions

- override void [InitDriftCompensation](#) ()  
*Initialise the drift compensation.*
- override int [GetFixationFrameCount](#) (int durationThreshold)  
*Get the number of required gaze samples to compute a fixation. This is based on the duration threshold and the sample rate of the device.*
- override Vector3 [GetUnitDirection](#) ()  
*Get the unit vector pointing in the direction of the gaze vector.*

## Additional Inherited Members

### 8.31.1 Detailed Description

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

See also

[GazeHelper.TrackerHandler](#)

### 8.31.2 Constructor & Destructor Documentation

#### 8.31.2.1 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>config</i>	The config item.

### 8.31.3 Member Function Documentation

#### 8.31.3.1 ApplyCalibration()

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

Compute and apply the calibration data. Transform the Tobii calibration result into the [GazeCalibrationData](#) structure.

**Returns**

The calibration data result wrapped by an async handler.

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

**8.31.3.2 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](#).

**8.31.3.3 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](#).

#### 8.31.3.4 ComputeValidation()

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

Compute the validation data.

##### Returns

The validation data result.

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

#### 8.31.3.5 FinishCalibration()

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

Finish the screen based calibration process.

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

#### 8.31.3.6 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](#).

#### 8.31.3.7 FinishValidation()

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

Finish the screen based validation process.

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

#### 8.31.3.8 GetFixationFrameCount()

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

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



## Parameters

<i>durationThreshold</i>	The required fixation duration in milliseconds.
--------------------------	---

## Returns

The number of required samples.

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

**8.31.3.9 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](#).

**8.31.3.10 InitCalibration()**

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

Initialise the screen based calibration.

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

**8.31.3.11 InitCalibrationAsync()**

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

Initialise the screen based calibration.

## Returns

An async handler

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

### 8.31.3.12 InitDriftCompensation()

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

Initialise the drift compensation.

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

### 8.31.3.13 InitValidation()

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

Initialise the screen based calibration.

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

### 8.31.3.14 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](#).

### 8.31.3.15 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.

### 8.31.3.16 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](#).

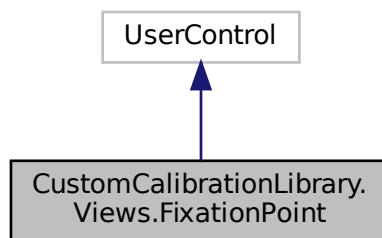
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Tracker/EyeTrackerPro.cs

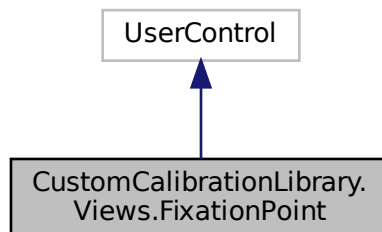
## 8.32 CustomCalibrationLibrary.Views.FixationPoint Class Reference

Interaction logic for FixationPoint.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.FixationPoint:



Collaboration diagram for CustomCalibrationLibrary.Views.FixationPoint:



## Public Member Functions

- [FixationPoint](#) ()

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

### 8.32.1 Detailed Description

Interaction logic for FixationPoint.xaml

### 8.32.2 Constructor & Destructor Documentation

#### 8.32.2.1 FixationPoint()

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

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

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

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

## 8.33 GazeUtilityLibrary.DataStructs.GazeCalibrationData Class Reference

The gaze calibration data structure

## 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]  
*The x coordinate of the calibration point.*
- double **YCoord** [get]  
*The y coordinate of the calibration point.*
- double **XCoordLeft** [get]  
*The x coord of the gaze point of the left eye.*
- double **YCoordLeft** [get]  
*The y coord of the gaze point of the left eye.*
- bool **ValidityLeft** [get]  
*The validity of gaze point coordinate of the left eye.*
- double **XCoordRight** [get]  
*The x coord of the gaze point of the right eye.*
- double **YCoordRight** [get]  
*The y coord of the gaze point of the right eye.*
- bool **ValidityRight** [get]  
*The validity of gaze point coordinate of the right eye.*

### 8.33.1 Detailed Description

The gaze calibration data structure

### 8.33.2 Constructor & Destructor Documentation

#### 8.33.2.1 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.

### 8.33.3 Member Function Documentation

#### 8.33.3.1 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.

### 8.33.4 Property Documentation

#### 8.33.4.1 ValidityLeft

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

The validity of gaze point coordinate of the left eye.

#### 8.33.4.2 ValidityRight

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

The validity of gaze point coordinate of the right eye.

#### 8.33.4.3 XCoord

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

The x coordinate of the calibration point.

#### 8.33.4.4 XCoordLeft

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

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

#### 8.33.4.5 XCoordRight

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

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

#### 8.33.4.6 YCoord

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

The y coordinate of the calibration point.

#### 8.33.4.7 YCoordLeft

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

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

#### 8.33.4.8 YCoordRight

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

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

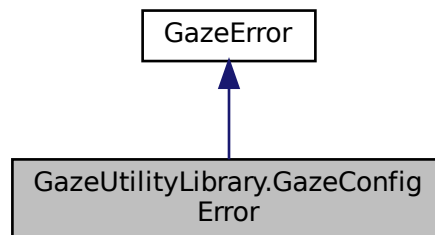
The documentation for this class was generated from the following file:

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

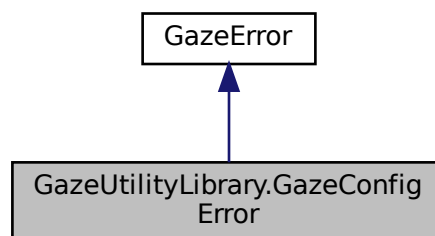
## 8.34 GazeUtilityLibrary.GazeConfigError Class Reference

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

Inheritance diagram for GazeUtilityLibrary.GazeConfigError:



Collaboration diagram for GazeUtilityLibrary.GazeConfigError:



### Public Member Functions

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

### Properties

- [EGazeConfigError Error](#) [set]  
*The error flags.*



## Additional Inherited Members

### 8.34.1 Detailed Description

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

### 8.34.2 Member Function Documentation

#### 8.34.2.1 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

### 8.34.3 Property Documentation

#### 8.34.3.1 Error

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

The error flags.

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

- source/GazeUtilityLibrary/GazeError.cs

## 8.35 GazeUtilityLibrary.GazeConfiguration Class Reference

The gaze configuration handler.

## Public Member Functions

- [GazeConfiguration](#) ([TrackerLogger](#) logger)  
*Initializes a new instance of the [GazeConfiguration](#) class.*
- bool [InitConfig](#) ()  
*Initialise the gaze configuration by parsing and checking the configuration file.*
- bool [CleanupGazeOutputFile](#) (string error)  
*Close the gaze outputfile and rename it by appending error codes.*
- bool [CleanupCalibrationOutputFile](#) (string error)  
*Close the calibration outputfile and rename it by appending error codes.*
- bool [CleanupValidationOutputFile](#) (string error)  
*Close the validation outputfile and rename it by appending error codes.*
- bool [DumpCurrentConfigurationFile](#) ()  
*Dump current configuration to the disk.*
- bool [PrepareGazeOutputFile](#) (string? subjectCode, string? outputPath)  
*Prepare the gaze output file based on the configuration.*
- bool [PrepareCalibrationOutputFile](#) (string? subjectCode)  
*Prepare the calibration output file based on the configuration.*
- bool [PrepareValidationOutputFile](#) (string? subjectCode)  
*Prepare the validation output file based on the configuration.*
- void [WriteToGazeOutput](#) (string[] formatted\_values)  
*Write to the gaze output file*
- void [WriteToCalibrationOutput](#) (string[] formatted\_values)  
*Write to the calibration output file*
- void [WriteToValidationOutput](#) (string[] formatted\_values)  
*Write to the calibration output file*

## Properties

- [ConfigItem?? Config](#) [get]  
*The JSON structure holding the configuratyon options.*

### 8.35.1 Detailed Description

The gaze configuration handler.

### 8.35.2 Constructor & Destructor Documentation

#### 8.35.2.1 GazeConfiguration()

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

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

## Parameters

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

### 8.35.3 Member Function Documentation

#### 8.35.3.1 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.

#### 8.35.3.2 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.

#### 8.35.3.3 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.

**8.35.3.4 DumpCurrentConfigurationFile()**

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

Dump current configuration to the disk.

**Returns**

True on success, False on failure.

**8.35.3.5 InitConfig()**

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

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

**Returns**

True on success, False on failure.

**8.35.3.6 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.

**8.35.3.7 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.

**8.35.3.8 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.

**8.35.3.9 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.
-------------------------	---

**8.35.3.10 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.
-------------------------	---

**8.35.3.11 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.
-------------------------	---

**8.35.4 Property Documentation****8.35.4.1 Config**

```
ConfigItem?? GazeUtilityLibrary.GazeConfiguration.Config [get]
```

The JSON structure holding the configuratyion options.

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

- source/GazeUtilityLibrary/GazeConfiguration.cs

## 8.36 GazeUtilityLibrary.DataStructs.GazeData Class Reference

The class definition of a gaze data set

### Public Member Functions

- [GazeData](#) (TimeSpan timestamp, TimeSpan timestampReceived, Vector2 gazePoint2d, bool isGazePoint2dValid)  
*Initializes a new instance of the GazeDataArgs class.*
- [GazeData](#) (TimeSpan timestamp, TimeSpan timestampReceived, Vector2 gazePoint2dLeft, bool isGazePoint2dValidLeft, Vector2 gazePoint2dRight, bool isGazePoint2dValidRight)  
*Initializes a new instance of the GazeDataArgs class.*
- [GazeData](#) (TimeSpan timestamp, TimeSpan timestampReceived, Vector2 gazePoint2dLeft, bool isGazePoint2dValidLeft, Vector2 gazePoint2dRight, bool isGazePoint2dValidRight, Vector3 gazePoint3dLeft, bool isGazePoint3dValidLeft, Vector3 gazePoint3dRight, bool isGazePoint3dValidRight, Vector3 gazeOrigin3dLeft, bool isGazeOrigin3dValidLeft, Vector3 gazeOrigin3dRight, bool isGazeOrigin3dValidRight, float pupilDiameterLeft, bool isPupilDiameterValidLeft, float pupilDiameterRight, bool isPupilDiameterValidRight)  
*Initializes a new instance of the GazeDataArgs class.*
- string[] [Prepare](#) ([ConfigItem](#) config, int trialId, string tag, TimeSpan startTime)  
*Prepare a list of formatted gaze data values*

### Properties

- TimeSpan [Timestamp](#) [get]  
*The timestamp of the data sample.*
- TimeSpan [TimestampReceived](#) [get]  
*The device timestamp of the data sample.*
- [GazeDataCollection? Left](#) [get]  
*The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data of the left eye.*
- [GazeDataCollection? Right](#) [get]  
*The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data of the right eye.*
- [GazeDataCollection Combined](#) [get]  
*The gaze data set, including 2d and (optionally) 3d gaze data as well as optional eye data of the combined eyes.*
- [DriftCompensationData? DriftCompensation](#) [get, set]  
*The drift compensation information.*

### 8.36.1 Detailed Description

The class definition of a gaze data set

### 8.36.2 Constructor & Destructor Documentation

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

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

Initializes a new instance of the GazeDataArgs class.

## Parameters

<i>timestamp</i>	The timestamp when the data was captured by the device.
<i>timestampReceived</i>	The timestamp when the data was received by the system.
<i>gazePoint2d</i>	The 2d coordinates of the combined gaze point.
<i>isGazePoint2dValid</i>	The validity of the combined 2d gaze point.

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

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

Initializes a new instance of the GazeDataArgs class.

## Parameters

<i>timestamp</i>	The timestamp when the data was captured by the device.
<i>timestampReceived</i>	The timestamp when the data was received by the system.
<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.

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

```
GazeUtilityLibrary.DataStructs.GazeData.GazeData (
    TimeSpan timestamp,
    TimeSpan timestampReceived,
    Vector2 gazePoint2dLeft,
    bool isGazePoint2dValidLeft,
    Vector2 gazePoint2dRight,
    bool isGazePoint2dValidRight,
    Vector3 gazePoint3dLeft,
    bool isGazePoint3dValidLeft,
    Vector3 gazePoint3dRight,
    bool isGazePoint3dValidRight,
    Vector3 gazeOrigin3dLeft,
    bool isGazeOrigin3dValidLeft,
    Vector3 gazeOrigin3dRight,
    bool isGazeOrigin3dValidRight,
    float pupilDiameterLeft,
```



```

bool isPupilDiameterValidLeft,
float pupilDiameterRight,
bool isPupilDiameterValidRight ) [inline]

```

Initializes a new instance of the GazeDataArgs class.

#### Parameters

<i>timestamp</i>	The timestamp when the data was captured by the device.
<i>timestampReceived</i>	The timestamp when the data was received by the system.
<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.

## 8.36.3 Member Function Documentation

### 8.36.3.1 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.

## 8.36.4 Property Documentation

### 8.36.4.1 Combined

`GazeDataCollection` `GazeUtilityLibrary.DataStructs.GazeData.Combined` [get]

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

### 8.36.4.2 DriftCompensation

`DriftCompensationData?` `GazeUtilityLibrary.DataStructs.GazeData.DriftCompensation` [get], [set]

The drift compensation information.

### 8.36.4.3 Left

`GazeDataCollection?` `GazeUtilityLibrary.DataStructs.GazeData.Left` [get]

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

### 8.36.4.4 Right

`GazeDataCollection?` `GazeUtilityLibrary.DataStructs.GazeData.Right` [get]

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

### 8.36.4.5 Timestamp

`TimeSpan` `GazeUtilityLibrary.DataStructs.GazeData.Timestamp` [get]

The timestamp of the data sample.

#### 8.36.4.6 TimestampReceived

TimeSpan GazeUtilityLibrary.DataStructs.GazeData.TimestampReceived [get]

The device timestamp of the data sample.

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

- source/GazeUtilityLibrary/DataStructs/GazeData.cs

## 8.37 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]

*The 2d gaze point.*

- bool [IsGazePointValid](#) [get]

*The validity flag of the 2d gaze point.*

#### 8.37.1 Detailed Description

The 2d gaze data set.

#### 8.37.2 Constructor & Destructor Documentation

##### 8.37.2.1 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.

### 8.37.3 Property Documentation

#### 8.37.3.1 GazePoint

`Vector2 GazeUtilityLibrary.DataStructs.GazeData2d.GazePoint [get]`

The 2d gaze point.

#### 8.37.3.2 IsGazePointValid

`bool GazeUtilityLibrary.DataStructs.GazeData2d.IsGazePointValid [get]`

The validity flag of the 2d gaze point.

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

- `source/GazeUtilityLibrary/DataStructs/GazeData2d.cs`

## 8.38 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]  
*The 3d gaze point.*
- bool [IsGazePointValid](#) [get]  
*The validity of the 3d gaze point.*
- Vector3 [GazeOrigin](#) [get]  
*The 3d origin of the gaze.*
- Vector3 [GazeDirection](#) [get]  
*The 3d gaze direction vector.*
- float [GazeDistance](#) [get]  
*The gaze distance from the origin to the gaze point.*
- bool [IsGazeOriginValid](#) [get]  
*The validity of the 3d origin.*

### 8.38.1 Detailed Description

The 3d gaze data set.

### 8.38.2 Constructor & Destructor Documentation

#### 8.38.2.1 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.

### 8.38.3 Property Documentation

#### 8.38.3.1 GazeDirection

```
Vector3 GazeUtilityLibrary.DataStructs.GazeData3d.GazeDirection [get]
```

The 3d gaze direction vector.

#### 8.38.3.2 GazeDistance

```
float GazeUtilityLibrary.DataStructs.GazeData3d.GazeDistance [get]
```

The gaze distance from the origin to the gaze point.

### 8.38.3.3 GazeOrigin

```
Vector3 GazeUtilityLibrary.DataStructs.GazeData3d.GazeOrigin [get]
```

The 3d origin of the gaze.

### 8.38.3.4 GazePoint

```
Vector3 GazeUtilityLibrary.DataStructs.GazeData3d.GazePoint [get]
```

The 3d gaze point.

### 8.38.3.5 IsGazeOriginValid

```
bool GazeUtilityLibrary.DataStructs.GazeData3d.IsGazeOriginValid [get]
```

The validity of the 3d origin.

### 8.38.3.6 IsGazePointValid

```
bool GazeUtilityLibrary.DataStructs.GazeData3d.IsGazePointValid [get]
```

The validity of the 3d gaze point.

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

- `source/GazeUtilityLibrary/DataStructs/GazeData3d.cs`

## 8.39 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]  
*The 2d gaze data.*
- [GazeData3d?](#) [GazeData3d](#) [get]  
*The 3d gaze data.*
- [EyeData?](#) [EyeData](#) [get]  
*Pupil data of the eye.*

### 8.39.1 Detailed Description

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

### 8.39.2 Constructor & Destructor Documentation

#### 8.39.2.1 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.

#### 8.39.2.2 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.

### 8.39.3 Property Documentation

#### 8.39.3.1 EyeData

`EyeData?` `GazeUtilityLibrary.DataStructs.GazeDataCollection.EyeData` [get]

Pupil data of the eye.

#### 8.39.3.2 GazeData2d

`GazeData2d` `GazeUtilityLibrary.DataStructs.GazeDataCollection.GazeData2d` [get]

The 2d gaze data.

#### 8.39.3.3 GazeData3d

`GazeData3d?` `GazeUtilityLibrary.DataStructs.GazeDataCollection.GazeData3d` [get]

The 3d gaze data.

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

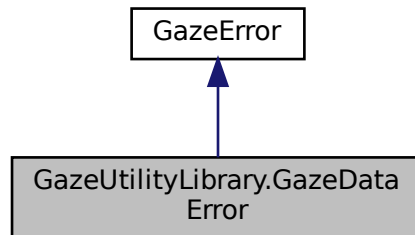
- `source/GazeUtilityLibrary/DataStructs/GazeDataCollection.cs`



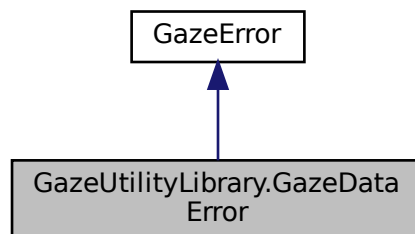
## 8.40 GazeUtilityLibrary.GazeDataError Class Reference

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

Inheritance diagram for GazeUtilityLibrary.GazeDataError:



Collaboration diagram for GazeUtilityLibrary.GazeDataError:



### Public Member Functions

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

### Properties

- [EGazeDataError Error](#) [set]  
*The error flags.*

## Additional Inherited Members

### 8.40.1 Detailed Description

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

### 8.40.2 Member Function Documentation

#### 8.40.2.1 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

### 8.40.3 Property Documentation

#### 8.40.3.1 Error

```
EGazeDataError GazeUtilityLibrary.GazeDataError.Error [set]
```

The error flags.

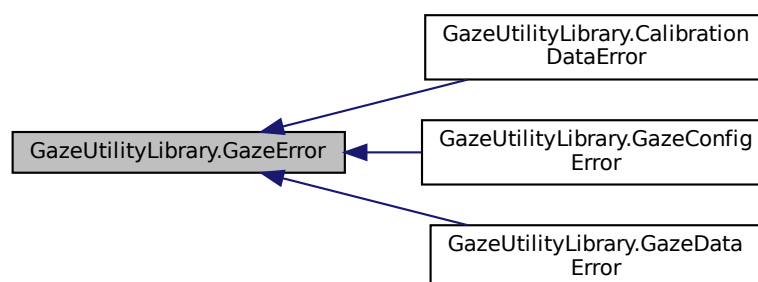
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/GazeError.cs

## 8.41 GazeUtilityLibrary.GazeError Class Reference

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

Inheritance diagram for GazeUtilityLibrary.GazeError:



## Protected Member Functions

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

### 8.41.1 Detailed Description

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

### 8.41.2 Member Function Documentation

#### 8.41.2.1 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

## 8.42 GazeUtilityLibrary.DataStructs.GazeValidationData Class Reference

The gaze validation data structure

## Public Member Functions

- [GazeValidationData](#) ()  
*Initializes a new instance of the [GazeValidationData](#) class.*
- [GazeValidationData](#) (float accuracyLeft, float accuracyRight, float precisionLeft, float precisionRight, float precisionRmsLeft, float precisionRmsRight)  
*Initializes a new instance of the [GazeValidationData](#) class.*
- void [AddPoint](#) (Vector2 point, float accuracyLeft, float accuracyRight, float precisionLeft, float precisionRight, float precisionRmsLeft, float precisionRmsRight)  
*Add a new validation point to the list.*

## Properties

- float [AccuracyLeft](#) [get]  
*The accuracy in degrees averaged over all collected points for the left eye.*
- float [AccuracyRight](#) [get]  
*The accuracy in degrees averaged over all collected points for the right eye.*
- float [PrecisionLeft](#) [get]  
*The precision (standard deviation) in degrees averaged over all collected points for the left eye.*
- float [PrecisionRight](#) [get]  
*The precision (standard deviation) in degrees averaged over all collected points for the right eye.*
- float [PrecisionRmsLeft](#) [get]  
*The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the left eye.*
- float [PrecisionRmsRight](#) [get]  
*The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the right eye.*
- List< [GazeValidationPoint](#) > [Points](#) [get]  
*The list of all*

### 8.42.1 Detailed Description

The gaze validation data structure

### 8.42.2 Constructor & Destructor Documentation

#### 8.42.2.1 GazeValidationData() [1/2]

```
GazeUtilityLibrary.DataStructs.GazeValidationData.GazeValidationData ( ) [inline]
```

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

#### 8.42.2.2 GazeValidationData() [2/2]

```
GazeUtilityLibrary.DataStructs.GazeValidationData.GazeValidationData (
    float accuracyLeft,
    float accuracyRight,
    float precisionLeft,
    float precisionRight,
    float precisionRmsLeft,
    float precisionRmsRight ) [inline]
```

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

## Parameters

<i>accuracyLeft</i>	The accuracy in degrees averaged over all collected points for the left eye.
<i>accuracyRight</i>	The accuracy in degrees averaged over all collected points for the right eye.
<i>precisionLeft</i>	The precision (standard deviation) in degrees averaged over all collected points for the left eye.
<i>precisionRight</i>	The precision (standard deviation) in degrees averaged over all collected points for the right eye.
<i>precisionRmsLeft</i>	The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the left eye.
<i>precisionRmsRight</i>	The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the right eye.

### 8.42.3 Member Function Documentation

#### 8.42.3.1 AddPoint()

```
void GazeUtilityLibrary.DataStructs.GazeValidationData.AddPoint (
    Vector2 point,
    float accuracyLeft,
    float accuracyRight,
    float precisionLeft,
    float precisionRight,
    float precisionRmsLeft,
    float precisionRmsRight ) [inline]
```

Add a new validation point to the list.

## Parameters

<i>point</i>	The validation point coordinates.
<i>accuracyLeft</i>	The accuracy in degrees averaged over all collected points for the left eye.
<i>accuracyRight</i>	The accuracy in degrees averaged over all collected points for the right eye.
<i>precisionLeft</i>	The precision (standard deviation) in degrees averaged over all collected points for the left eye.
<i>precisionRight</i>	The precision (standard deviation) in degrees averaged over all collected points for the right eye.
<i>precisionRmsLeft</i>	The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the left eye.
<i>precisionRmsRight</i>	The precision (root mean square of sample-to-sample error) in degrees averaged over all collected points for the right eye.

### 8.42.4 Property Documentation

#### 8.42.4.1 AccuracyLeft

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.AccuracyLeft [get]
```

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

#### 8.42.4.2 AccuracyRight

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.AccuracyRight [get]
```

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

#### 8.42.4.3 Points

```
List<GazeValidationPoint> GazeUtilityLibrary.DataStructs.GazeValidationData.Points [get]
```

The list of all

#### 8.42.4.4 PrecisionLeft

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.PrecisionLeft [get]
```

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

#### 8.42.4.5 PrecisionRight

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.PrecisionRight [get]
```

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

#### 8.42.4.6 PrecisionRmsLeft

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.PrecisionRmsLeft [get]
```

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

#### 8.42.4.7 PrecisionRmsRight

```
float GazeUtilityLibrary.DataStructs.GazeValidationData.PrecisionRmsRight [get]
```

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

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

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

## 8.43 GazeUtilityLibrary.DataStructs.GazeValidationPoint Class Reference

A validation point.

### Public Member Functions

- [GazeValidationPoint](#) (Vector2 point, [GazeValidationData](#) result)  
*Initializes a new instance of the [GazeValidationPoint](#) class.*
- string[] [Prepare](#) ([ConfigItem](#) config)  
*Prepare a list of formatted calibration data values*

### Properties

- Vector2 [Point](#) [get]  
*The validation point.*
- [GazeValidationData Result](#) [get]  
*The validation result of this point.*

#### 8.43.1 Detailed Description

A validation point.

#### 8.43.2 Constructor & Destructor Documentation

##### 8.43.2.1 GazeValidationPoint()

```
GazeUtilityLibrary.DataStructs.GazeValidationPoint.GazeValidationPoint (
    Vector2 point,
    GazeValidationData result ) [inline]
```

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

**Parameters**

<i>point</i>	The validation point.
<i>result</i>	The validation result of this point.

### 8.43.3 Member Function Documentation

#### 8.43.3.1 Prepare()

```
string [ ] GazeUtilityLibrary.DataStructs.GazeValidationPoint.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.

### 8.43.4 Property Documentation

#### 8.43.4.1 Point

```
Vector2 GazeUtilityLibrary.DataStructs.GazeValidationPoint.Point [get]
```

The validation point.

#### 8.43.4.2 Result

```
GazeValidationData GazeUtilityLibrary.DataStructs.GazeValidationPoint.Result [get]
```

The validation result of this point.

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

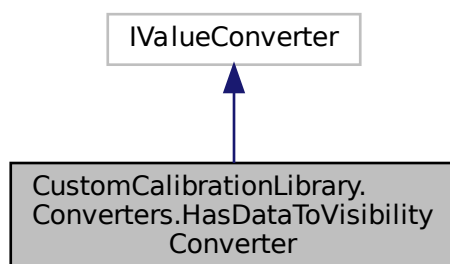
- source/GazeUtilityLibrary/DataStructs/GazeValidationData.cs



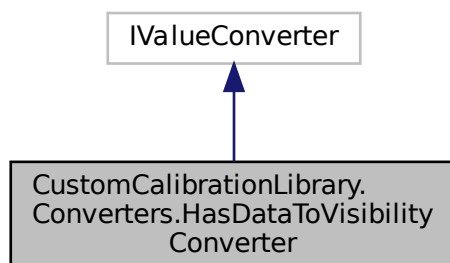
## 8.44 CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter Class Reference

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)  
*Value converter.*
- object [ConvertBack](#) (object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)  
*Reverted value converter.*

#### 8.44.1 Detailed Description

Converts True to Hidden and False to Visible

## 8.44.2 Member Function Documentation

### 8.44.2.1 Convert()

```
object CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter.Convert (
    object value,
    Type targetType,
    object parameter,
    System.Globalization.CultureInfo culture ) [inline]
```

Value converter.

#### Parameters

<i>value</i>	The value to convert.
<i>targetType</i>	The type of the target value.
<i>parameter</i>	The conversion parameter.
<i>culture</i>	The language localisation.

#### Returns

The converted value object

#### Exceptions

<i>InvalidOperationException</i>	
----------------------------------	--

### 8.44.2.2 ConvertBack()

```
object CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter.ConvertBack (
    object value,
    Type targetType,
    object parameter,
    System.Globalization.CultureInfo culture ) [inline]
```

Reverted value converter.

#### Parameters

<i>value</i>	The value to convert.
<i>targetType</i>	The type of the target value.
<i>parameter</i>	The conversion parameter.
<i>culture</i>	The language localisation.

**Returns**

The converted value object

**Exceptions**

<i>NotSupportedException</i>	
------------------------------	--

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

- source/CustomCalibrationLibrary/Converters/HasDataToVisibilityConverter.cs

## 8.45 GazeUtilityLibrary.JsonConfigParser Class Reference

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

### Public Member Functions

- [JsonConfigParser](#) ([TrackerLogger](#) logger)  
*Initializes a new instance of the [JsonConfigParser](#) class.*
- [ConfigItem?](#) [ParseJsonConfig](#) (ref [GazeConfigError](#) error)  
*Parses the json configuration.*
- void [SerializeJsonConfig](#) ([ConfigItem](#) item, string path)  
*Serializes the json configuration object to a string and writes it to a file.*
- [ConfigItem](#) [GetDefaultConfig](#) ()  
*Gets the default configuration values.*

### 8.45.1 Detailed Description

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

### 8.45.2 Constructor & Destructor Documentation

#### 8.45.2.1 JsonConfigParser()

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

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

**Parameters**

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

### 8.45.3 Member Function Documentation

#### 8.45.3.1 GetDefaultConfig()

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

Gets the default configuration values.

##### Returns

the default configuration values.

#### 8.45.3.2 ParseJsonConfig()

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

Parses the json configuration.

##### Returns

the updated [ConfigItem](#) class.

#### 8.45.3.3 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.

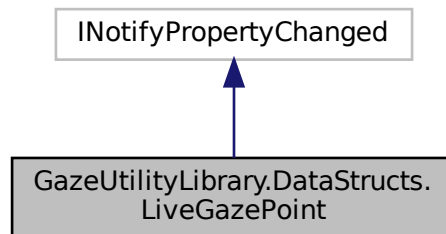
The documentation for this class was generated from the following file:

- `source/GazeUtilityLibrary/GazeConfiguration.cs`

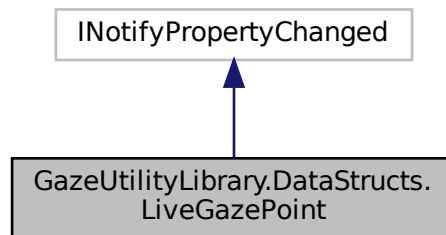
## 8.46 GazeUtilityLibrary.DataStructs.LiveGazePoint Class Reference

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

Inheritance diagram for GazeUtilityLibrary.DataStructs.LiveGazePoint:



Collaboration diagram for GazeUtilityLibrary.DataStructs.LiveGazePoint:



### Properties

- double `X` [get, set]  
*The normalized x coordinate on the screen*
- double `Y` [get, set]  
*The normalized y coordinate on the screen*
- bool `Visibility` [get, set]  
*The visibility flag.*

### Events

- PropertyChangedEventHandler? `PropertyChanged`  
*Event to trigger property changes.*

### 8.46.1 Detailed Description

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

### 8.46.2 Property Documentation

#### 8.46.2.1 Visibility

```
bool GazeUtilityLibrary.DataStructs.LiveGazePoint.Visibility [get], [set]
```

The visibility flag.

#### 8.46.2.2 X

```
double GazeUtilityLibrary.DataStructs.LiveGazePoint.X [get], [set]
```

The normalized x coordinate on the screen

#### 8.46.2.3 Y

```
double GazeUtilityLibrary.DataStructs.LiveGazePoint.Y [get], [set]
```

The normalized y coordinate on the screen

### 8.46.3 Event Documentation

#### 8.46.3.1 PropertyChanged

```
PropertyChangedEventHandler? GazeUtilityLibrary.DataStructs.LiveGazePoint.PropertyChanged
```

Event to trigger property changes.

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

- source/GazeUtilityLibrary/DataStructs/LiveGazePoint.cs

## 8.47 CustomCalibrationLibrary.ViewModels.Monitor Class Reference

A representation of the screen.

### Public Member Functions

- [Monitor](#) (int index, string name)  
*Initializes a new instance of the [Monitor](#) class.*

### Properties

- string [Name](#) [get]  
*The name of the screen.*
- int [Index](#) [get]  
*The screen index.*

#### 8.47.1 Detailed Description

A representation of the screen.

#### 8.47.2 Constructor & Destructor Documentation

##### 8.47.2.1 Monitor()

```
CustomCalibrationLibrary.ViewModels.Monitor.Monitor (  
    int index,  
    string name ) [inline]
```

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

##### Parameters

<i>index</i>	The screen index.
<i>name</i>	The name of the screen.

#### 8.47.3 Property Documentation

##### 8.47.3.1 Index

```
int CustomCalibrationLibrary.ViewModels.Monitor.Index [get]
```

The screen index.

### 8.47.3.2 Name

```
string CustomCalibrationLibrary.ViewModels.Monitor.Name [get]
```

The name of the screen.

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

- source/CustomCalibrationLibrary/ViewModels/ScreenSelectionViewModel.cs

## 8.48 GazeUtilityLibrary.MouseHider Class Reference

hide standard mouse pointer and restore it

### Public Member Functions

- [MouseHider](#) ([TrackerLogger](#) logger)  
*Initializes a new instance of the [MouseHider](#) class.*
- void [HideCursor](#) ()  
*Hides the cursor.*
- void [ShowCursor](#) (string? pathToCur)  
*Shows the cursor.*

### 8.48.1 Detailed Description

hide standard mouse pointer and restore it

### 8.48.2 Constructor & Destructor Documentation

#### 8.48.2.1 MouseHider()

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

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



## Parameters

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

### 8.48.3 Member Function Documentation

#### 8.48.3.1 HideCursor()

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

Hides the cursor.

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

#### 8.48.3.2 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.
------------------	--

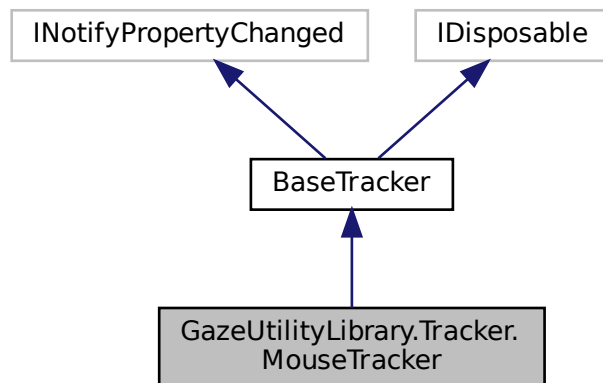
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/MouseHider.cs

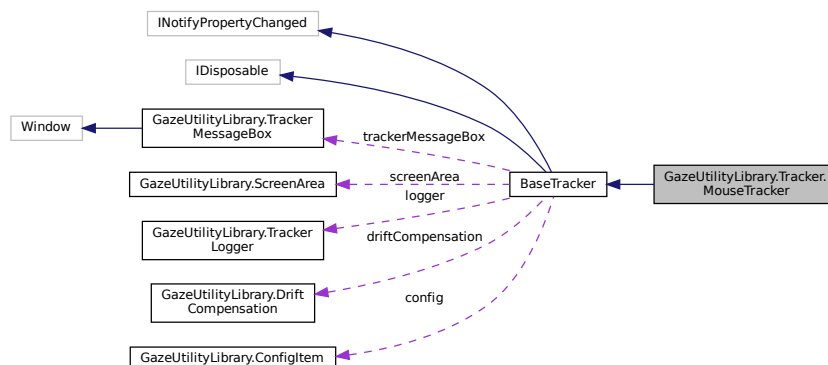
## 8.49 GazeUtilityLibrary.Tracker.MouseTracker Class Reference

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](#) (int durationThreshold)
  - Get the number of required gaze samples to compute a fixation. This is device specific and must be overwritten by the device because the duration of fixation point detection depends on the frame rate of the device.*
- override Vector3 [GetUnitDirection](#) ()
  - Get the unit vector pointing in the direction of the gaze vector. This is device specific as the gaze data are represented in a coordinate system as defined by the device.*
- override void [InitDriftCompensation](#) ()
  - Initialise the drift compensation. This is device specific and must be overwritten by the device class.*

## Additional Inherited Members

### 8.49.1 Detailed Description

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

See also

GazeHelper.TrackerHandler

### 8.49.2 Constructor & Destructor Documentation

#### 8.49.2.1 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>config</i>	The config item.

### 8.49.3 Member Function Documentation

#### 8.49.3.1 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](#).

#### 8.49.3.2 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](#).

#### 8.49.3.3 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](#).

**8.49.3.4 ComputeValidation()**

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

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

**Returns**

The validation data result.

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

**8.49.3.5 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](#).

**8.49.3.6 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](#).

### 8.49.3.7 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](#).

### 8.49.3.8 FinishValidation()

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

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

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

### 8.49.3.9 GetFixationFrameCount()

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

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

#### Parameters

<i>durationThreshold</i>	The required fixation duration in milliseconds.
--------------------------	---

#### Returns

The number of gaze samples to require for fixation detection.

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

### 8.49.3.10 GetUnitDirection()

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

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

**Returns**

The unit vector

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

**8.49.3.11 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](#).

**8.49.3.12 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](#).

**8.49.3.13 InitDriftCompensation()**

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

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

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

**8.49.3.14 InitValidation()**

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

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

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

### 8.49.3.15 Start()

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

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

### 8.49.3.16 Stop()

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

Removes to mouse event hook.

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

- source/GazeUtilityLibrary/Tracker/MouseTracker.cs

## 8.50 GazeUtilityLibrary.DataStructs.PipeCommand Class Reference

The JSON structure of a pipe command.

### Public Member Functions

- [PipeCommand](#) (string? command, bool reset, int? trialId, string? label)  
*Initializes a new instance of the [PipeCommand](#) class.*

### Properties

- string? [Command](#) [get, set]  
*The optional pipe command to be sent.*
- string? [Label](#) [get, set]  
*An optional label to annotate gaze data.*
- int? [TrialId](#) [get, set]  
*An optional trial ID to annotate gaze data.*
- bool? [ResetStartTime](#) [get, set]  
*An optional flag to indicate whether the relative timestamp should be reset.*

### 8.50.1 Detailed Description

The JSON structure of a pipe command.

### 8.50.2 Constructor & Destructor Documentation

#### 8.50.2.1 PipeCommand()

```
GazeUtilityLibrary.DataStructs.PipeCommand.PipeCommand (
    string? command,
    bool reset,
    int? trialId,
    string? label ) [inline]
```

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



## Parameters

<i>command</i>	The pipe command to be sent.
<i>reset</i>	A flag to indicate whether the relative timestamp should be reset.
<i>trialId</i>	An optional trial ID to annotate gaze data.
<i>label</i>	An optional label to annotate gaze data.

### 8.50.3 Property Documentation

#### 8.50.3.1 Command

```
string? GazeUtilityLibrary.DataStructs.PipeCommand.Command [get], [set]
```

The optional pipe command to be sent.

#### 8.50.3.2 Label

```
string? GazeUtilityLibrary.DataStructs.PipeCommand.Label [get], [set]
```

An optional label to annotate gaze data.

#### 8.50.3.3 ResetStartTime

```
bool? GazeUtilityLibrary.DataStructs.PipeCommand.ResetStartTime [get], [set]
```

An optional flag to indicate whether the relative timestamp should be reset.

#### 8.50.3.4 TrialId

```
int? GazeUtilityLibrary.DataStructs.PipeCommand.TrialId [get], [set]
```

An optional trial ID to annotate gaze data.

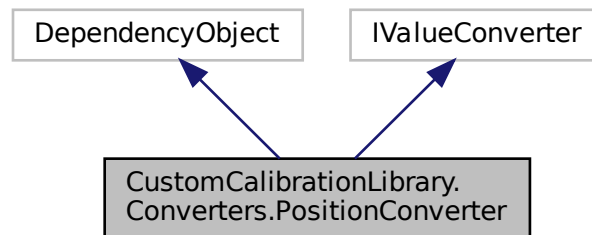
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/PipeCommand.cs

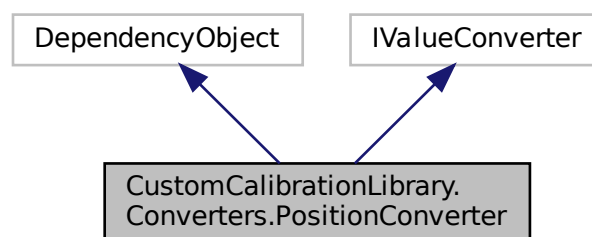
## 8.51 CustomCalibrationLibrary.Converters.PositionConverter Class Reference

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

Inheritance diagram for CustomCalibrationLibrary.Converters.PositionConverter:



Collaboration diagram for CustomCalibrationLibrary.Converters.PositionConverter:



### Public Member Functions

- object [Convert](#) (object value, Type targetType, object parameter, CultureInfo culture)  
*Value converter.*
- object [ConvertBack](#) (object value, Type targetType, object parameter, CultureInfo culture)  
*Reverted value converter.*

### Static Public Attributes

- static readonly DependencyProperty [OffsetProperty](#)  
*The custom offset property of the value converter.*

## Properties

- string?? [Offset](#) [get, set]  
*The position offset.*

### 8.51.1 Detailed Description

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

### 8.51.2 Member Function Documentation

#### 8.51.2.1 Convert()

```
object CustomCalibrationLibrary.Converters.PositionConverter.Convert (
    object value,
    Type targetType,
    object parameter,
    CultureInfo culture ) [inline]
```

Value converter.

#### Parameters

<i>value</i>	The value to convert.
<i>targetType</i>	The type of the target value.
<i>parameter</i>	The conversion parameter.
<i>culture</i>	The language localisation.

#### Returns

The converted value object

#### 8.51.2.2 ConvertBack()

```
object CustomCalibrationLibrary.Converters.PositionConverter.ConvertBack (
    object value,
    Type targetType,
    object parameter,
    CultureInfo culture ) [inline]
```

Reverted value converter.

**Parameters**

<i>value</i>	The value to convert.
<i>targetType</i>	The type of the target value.
<i>parameter</i>	The conversion parameter.
<i>culture</i>	The language localisation.

**Returns**

The converted value object

**Exceptions**

<i>NotSupportedException</i>	
------------------------------	--

**8.51.3 Member Data Documentation****8.51.3.1 OffsetProperty**

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

**Initial value:**

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

The custom offset property of the value converter.

**8.51.4 Property Documentation****8.51.4.1 Offset**

```
string?? CustomCalibrationLibrary.Converters.PositionConverter.Offset [get], [set]
```

The position offset.

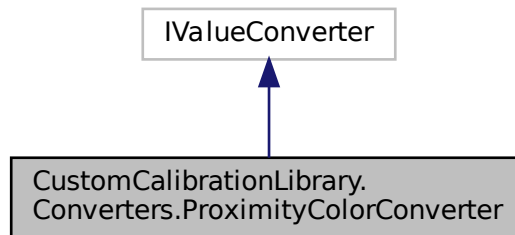
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/Converters/PositionConverter.cs

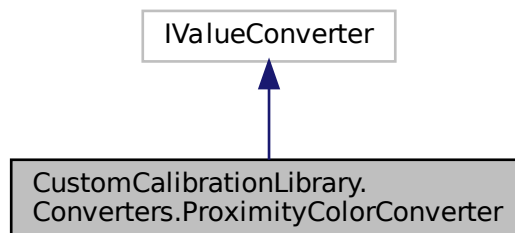
## 8.52 CustomCalibrationLibrary.Converters.ProximityColorConverter Class Reference

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

Inheritance diagram for CustomCalibrationLibrary.Converters.ProximityColorConverter:



Collaboration diagram for CustomCalibrationLibrary.Converters.ProximityColorConverter:



### Public Member Functions

- object [Convert](#) (object value, Type targetType, object parameter, CultureInfo culture)  
*Value converter.*
- object [ConvertBack](#) (object value, Type targetType, object parameter, CultureInfo culture)  
*Reverted value converter.*

#### 8.52.1 Detailed Description

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

## 8.52.2 Member Function Documentation

### 8.52.2.1 Convert()

```
object CustomCalibrationLibrary.Converters.ProximityColorConverter.Convert (
    object value,
    Type targetType,
    object parameter,
    CultureInfo culture ) [inline]
```

Value converter.

#### Parameters

<i>value</i>	The value to convert.
<i>targetType</i>	The type of the target value.
<i>parameter</i>	The conversion parameter.
<i>culture</i>	The language localisation.

#### Returns

The converted value object

### 8.52.2.2 ConvertBack()

```
object CustomCalibrationLibrary.Converters.ProximityColorConverter.ConvertBack (
    object value,
    Type targetType,
    object parameter,
    CultureInfo culture ) [inline]
```

Reverted value converter.

#### Parameters

<i>value</i>	The value to convert.
<i>targetType</i>	The type of the target value.
<i>parameter</i>	The conversion parameter.
<i>culture</i>	The language localisation.

#### Returns

The converted value object

## Exceptions

<code>NotSupportedException</code>	
------------------------------------	--

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

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

## 8.53 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]  
*The width of the screen.*
- float [Height](#) [get]  
*The height of the screen.*
- Vector3 [BottomLeft](#) [get]  
*The coordinates of the bottom left point of the screen.*
- Vector3 [BottomRight](#) [get]  
*The coordinates of the bottom right point of the screen.*
- Vector3 [TopLeft](#) [get]  
*The coordinates of the top left point of the screen.*
- Vector3 [TopRight](#) [get]  
*The coordinates of the top right point of the screen.*
- Vector3 [Center](#) [get]  
*The coordinates of the center point of the screen.*

### 8.53.1 Detailed Description

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

### 8.53.2 Constructor & Destructor Documentation

#### 8.53.2.1 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 height of the screen

### 8.53.3 Member Function Documentation

#### 8.53.3.1 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

### 8.53.3.2 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 screen 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.

### 8.53.3.3 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

### 8.53.3.4 GetPoint2dNormalized()

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

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.

**Parameters**

<i>point3d</i>	The 3d point on the screen plane to convert.
----------------	--

**Returns**

The normalized 2d point on the screen plane

## 8.53.4 Property Documentation

### 8.53.4.1 BottomLeft

`Vector3 GazeUtilityLibrary.ScreenArea.BottomLeft` [get]

The coordinates of the bottom left point of the screen.

### 8.53.4.2 BottomRight

`Vector3 GazeUtilityLibrary.ScreenArea.BottomRight` [get]

The coordinates of the bottom right point of the screen.

### 8.53.4.3 Center

`Vector3 GazeUtilityLibrary.ScreenArea.Center` [get]

The coordinates of the center point of the screen.

### 8.53.4.4 Height

`float GazeUtilityLibrary.ScreenArea.Height` [get]

The height of the screen.

#### 8.53.4.5 TopLeft

```
Vector3 GazeUtilityLibrary.ScreenArea.TopLeft [get]
```

The coordinates of the top left point of the screen.

#### 8.53.4.6 TopRight

```
Vector3 GazeUtilityLibrary.ScreenArea.TopRight [get]
```

The coordinates of the to right point of the screen.

#### 8.53.4.7 Width

```
float GazeUtilityLibrary.ScreenArea.Width [get]
```

The width of the screen.

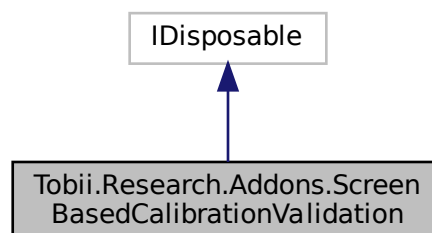
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/ScreenArea.cs

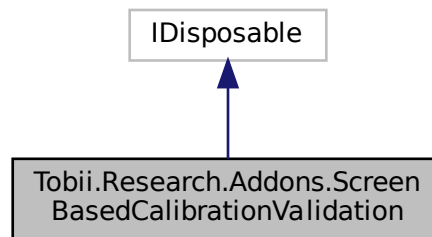
## 8.54 Tobii.Research.Addons.ScreenBasedCalibrationValidation Class Reference

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

Inheritance diagram for Tobii.Research.Addons.ScreenBasedCalibrationValidation:



Collaboration diagram for Tobii.Research.Addons.ScreenBasedCalibrationValidation:



## Public Types

- enum [ValidationState](#) { **NotInValidationMode**, **NotCollectingData**, **CollectingData** }

*ValidationState.NotInValidationMode* - [EnterValidationMode](#) must be called starting to collect data. *ValidationState.*↔  
*NotCollectingData* - Ready to start collecting data or computing result. *ValidationState.CollectingData* - Currently collecting data. Will finish after the sample count is reached or a timeout.

## Public Member Functions

- [ScreenBasedCalibrationValidation](#) (IEyeTracker eyeTracker, int sampleCount=30, int timeoutMS=1000)  
 Create a calibration validation object for screen based eye trackers.
- void [StartCollectingData](#) (NormalizedPoint2D calibrationPointCoordinates)  
 Starts collecting data for a calibration validation point. The argument used is the point the user is assumed to be looking at and is given in the active display area coordinate system. Please check State property to know when data collection is completed (or timed out).
- void [DiscardData](#) (NormalizedPoint2D calibrationPointCoordinates)  
 Removes the collected data for a specific calibration validation point.
- void [EnterValidationMode](#) ()  
 Enter the calibration validation mode and starts subscribing to gaze data from the eye tracker.
- void [LeaveValidationMode](#) ()  
 Leaves the calibration validation mode, clears all collected data, and unsubscribes from the eye tracker.
- [CalibrationValidationResult Compute](#) ()  
 Uses the collected data and tries to compute accuracy and precision values for all points. If the calculation is successful, the result is returned, and stored in the Result property of the CalibrationValidation object. If there is insufficient data to compute the results for a certain point that [CalibrationValidationPoint](#) will contain invalid data (NaN) for the results. Gaze data will still be untouched. If there is no valid data for any point, the average results of [CalibrationValidationResult](#) will be invalid (NaN) as well.
- void [Dispose](#) ()  
 Dispose will unsubscribe to gaze data and exit validation mode, if the object is not already in *ValidationState.NotInValidationMode*↔
- override string [ToString](#) ()  
 Convert validation values to a string.

## Properties

- [ValidationState State](#) [get]  
Get the current state of the validation object.
- [CalibrationValidationResult Result](#) [get]  
Get the current [CalibrationValidationResult](#) with the computed accuracy and precision. [Compute](#) must have been called for this to contain valid data.

### 8.54.1 Detailed Description

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

### 8.54.2 Member Enumeration Documentation

#### 8.54.2.1 ValidationState

```
enum Tobii.Research.Addons.ScreenBasedCalibrationValidation.ValidationState [strong]
```

[ValidationState.NotInValidationMode](#) - [EnterValidationMode](#) must be called starting to collect data. [ValidationState.NotCollectingData](#) - Ready to start collecting data or computing result. [ValidationState.CollectingData](#) - Currently collecting data. Will finish after the sample count is reached or a timeout.

### 8.54.3 Constructor & Destructor Documentation

#### 8.54.3.1 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 <a href="#">IEyeTracker</a> 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.

## 8.54.4 Member Function Documentation

### 8.54.4.1 Compute()

```
CalibrationValidationResult Tobii.Research.Addons.ScreenBasedCalibrationValidation.Compute ( )
[inline]
```

Uses the collected data and tries to compute accuracy and precision values for all points. If the calculation is successful, the result is returned, and stored in the Result property of the CalibrationValidation object. If there is insufficient data to compute the results for a certain point that [CalibrationValidationPoint](#) will contain invalid data (NaN) for the results. Gaze data will still be untouched. If there is no valid data for any point, the average results of [CalibrationValidationResult](#) will be invalid (NaN) as well.

#### Returns

The [CalibrationValidationResult](#)

### 8.54.4.2 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.
------------------------------------	----------------------------------

### 8.54.4.3 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

### 8.54.4.4 EnterValidationMode()

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

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

#### 8.54.4.5 LeaveValidationMode()

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

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

#### 8.54.4.6 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
------------------------------------	---

#### 8.54.4.7 ToString()

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

Convert validation values to a string.

##### Returns

The validation string.

### 8.54.5 Property Documentation

#### 8.54.5.1 Result

```
CalibrationValidationResult Tobii.Research.Addons.ScreenBasedCalibrationValidation.Result
[get]
```

Get the current [CalibrationValidationResult](#) with the computed accuracy and precision. [Compute](#) must have been called for this to contain valid data.



### 8.54.5.2 State

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

Get the current state of the validation object.

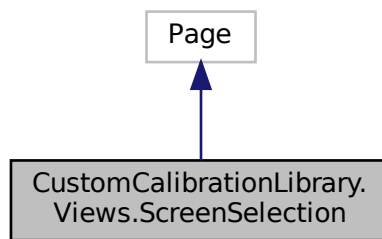
The documentation for this class was generated from the following file:

- source/TobiiProSdkAddons/ScreenBasedCalibrationValidation.cs

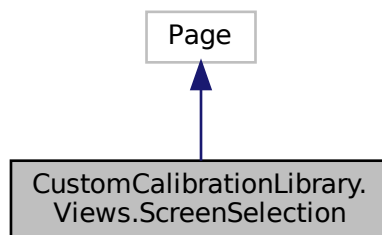
## 8.55 CustomCalibrationLibrary.Views.ScreenSelection Class Reference

Interaction logic for ScreenSelection.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.ScreenSelection:



Collaboration diagram for CustomCalibrationLibrary.Views.ScreenSelection:



### Public Member Functions

- `ScreenSelection` (`CalibrationModel` model, Window window)  
*Initializes a new instance of the `ScreenSelection` class.*

### 8.55.1 Detailed Description

Interaction logic for ScreenSelection.xaml

### 8.55.2 Constructor & Destructor Documentation

#### 8.55.2.1 ScreenSelection()

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

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

##### Parameters

<i>model</i>	The calibration model.
<i>window</i>	The target window.

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

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

## 8.56 CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel Class Reference

The view model class for the screen selection view.

### Public Member Functions

- [ScreenSelectionViewModel](#) ([CalibrationModel](#) model, [Window](#) window)  
*Initializes a new instance of the [ScreenSelectionViewModel](#) class.*

### Properties

- [ObservableCollection< \[Monitor\]\(#\) > \[Monitors\]\(#\)](#) [get]  
*The observable list of monitors to select from.*
- [ICommand](#) [CalibrationStartCommand](#) [get]  
*Command to start the calibration*
- [ICommand](#) [CalibrationAbortCommand](#) [get]  
*Command to abort the calibration*
- [ICommand](#) [ScreenSwitchCommand](#) [get]  
*Command to switch the screen*

### 8.56.1 Detailed Description

The view model class for the screen selection view.

### 8.56.2 Constructor & Destructor Documentation

#### 8.56.2.1 ScreenSelectionViewModel()

```
CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.ScreenSelectionViewModel (
    CalibrationModel model,
    Window window ) [inline]
```

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

##### Parameters

<i>model</i>	The calibration model
<i>window</i>	The target window of the screen selection

### 8.56.3 Property Documentation

#### 8.56.3.1 CalibrationAbortCommand

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

Command to abort the calibration

#### 8.56.3.2 CalibrationStartCommand

```
ICommand CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.CalibrationStartCommand
[get]
```

Command to start the calibration

### 8.56.3.3 Monitors

```
ObservableCollection<Monitor> CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.↔
Monitors [get]
```

The observable list of monitors to select from.

### 8.56.3.4 ScreenSwitchCommand

```
 ICommand CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel.ScreenSwitchCommand
[get]
```

Command to switch the screen

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

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

## 8.57 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 whether the gaze point is outside the screen area.

### Public Member Functions

- [ScreenTriangle](#) (Vector3 v1, Vector3 v2, Vector3 v3)  
*Initializes a new instance of the [ScreenTriangle](#) class.*
- Vector3? [GetIntersectionPoint](#) (Vector3 origin, Vector3 direction)  
*Compute the intersection point with the triangle with the Moller-Trumbore algorithm.*

### Properties

- Vector3 [V1](#) [get]  
*A corner point of the triangle.*
- Vector3 [V2](#) [get]  
*A corner point of the triangle.*
- Vector3 [V3](#) [get]  
*A corner point of the triangle.*
- Vector3 [E1](#) [get]  
*The edge vector from v1 to v2.*
- Vector3 [E2](#) [get]  
*The edge vector from v1 to v3.*

### 8.57.1 Detailed Description

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

### 8.57.2 Constructor & Destructor Documentation

#### 8.57.2.1 ScreenTriangle()

```
GazeUtilityLibrary.ScreenTriangle.ScreenTriangle (
    Vector3 v1,
    Vector3 v2,
    Vector3 v3 ) [inline]
```

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

##### Parameters

<i>v1</i>	A corner point of the triangle.
<i>v2</i>	A corner point of the triangle.
<i>v3</i>	A corner point of the triangle.

### 8.57.3 Member Function Documentation

#### 8.57.3.1 GetIntersectionPoint()

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

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

##### Parameters

<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.

## 8.57.4 Property Documentation

### 8.57.4.1 E1

`Vector3 GazeUtilityLibrary.ScreenTriangle.E1 [get]`

The edge vector from v1 to v2.

### 8.57.4.2 E2

`Vector3 GazeUtilityLibrary.ScreenTriangle.E2 [get]`

The edge vector from v1 to v3.

### 8.57.4.3 V1

`Vector3 GazeUtilityLibrary.ScreenTriangle.V1 [get]`

A corner point of the triangle.

### 8.57.4.4 V2

`Vector3 GazeUtilityLibrary.ScreenTriangle.V2 [get]`

A corner point of the triangle.

### 8.57.4.5 V3

`Vector3 GazeUtilityLibrary.ScreenTriangle.V3 [get]`

A corner point of the triangle.

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

- `source/GazeUtilityLibrary/ScreenTriangle.cs`

## 8.58 GazeUtilityLibrary.TrackerLogger Class Reference

Simple logger class.

### Public Member Functions

- [TrackerLogger](#) (string? logPath, [EOutputType](#) type=EOutputType.gaze)  
*Initializes a new instance of the [TrackerLogger](#) class.*
- void [DumpFatal](#) (Exception e)  
*Dumps exception to a new file if it is not possible to write to the main log file.*
- void [Debug](#) (string message)  
*wrapper function for debug level logging.*
- void [Info](#) (string message)  
*wrapper function for info level logging*
- void [Warning](#) (string message)  
*wrapper function for warning level logging*
- void [Error](#) (string message)  
*wrapper function for error level logging*

### 8.58.1 Detailed Description

Simple logger class.

### 8.58.2 Constructor & Destructor Documentation

#### 8.58.2.1 TrackerLogger()

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

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

### 8.58.3 Member Function Documentation

#### 8.58.3.1 Debug()

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

wrapper function for debug level logging.

## Parameters

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

**8.58.3.2 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.
----------	----------------

**8.58.3.3 Error()**

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

wrapper function for error level logging

## Parameters

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

**8.58.3.4 Info()**

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

wrapper function for info level logging

## Parameters

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



### 8.58.3.5 Warning()

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

wrapper function for warning level logging

#### Parameters

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

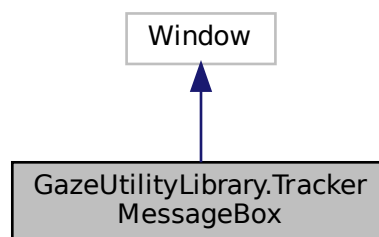
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/Logger.cs

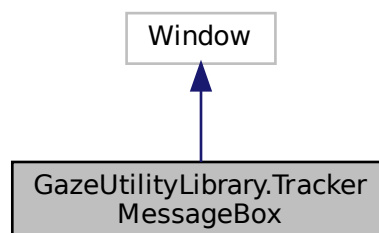
## 8.59 GazeUtilityLibrary.TrackerMessageBox Class Reference

Interaction logic for TrackerMessageBox.xaml

Inheritance diagram for GazeUtilityLibrary.TrackerMessageBox:



Collaboration diagram for GazeUtilityLibrary.TrackerMessageBox:



### 8.59.1 Detailed Description

Interaction logic for TrackerMessageBox.xaml

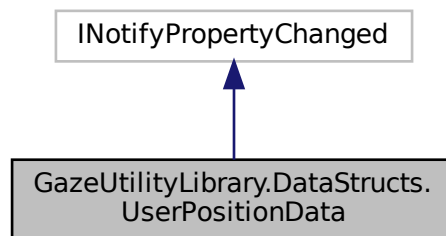
The documentation for this class was generated from the following file:

- `source/GazeUtilityLibrary/TrackerMessageBox.xaml.cs`

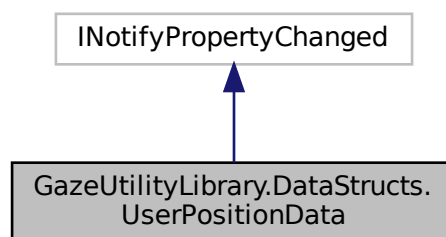
## 8.60 GazeUtilityLibrary.DataStructs.UserPositionData Class Reference

The user position to be rendered on the screen.

Inheritance diagram for GazeUtilityLibrary.DataStructs.UserPositionData:



Collaboration diagram for GazeUtilityLibrary.DataStructs.UserPositionData:



### Public Member Functions

- [UserPositionData](#) ()  
*Initializes a new instance of the [UserPositionData](#) class.*
- [UserPositionData](#) (double xCoordLeft, double yCoordLeft, double zCoordLeft, double xCoordRight, double yCoordRight, double zCoordRight)  
*Initializes a new instance of the [UserPositionData](#) class.*

## Properties

- double [XCoordLeft](#) [get, set]  
*The normalized x coordinate of the left eye.*
- double [YCoordLeft](#) [get, set]  
*The normalized y coordinate of the left eye.*
- double [ZCoordLeft](#) [get, set]  
*The normalized z coordinate of the left eye.*
- double [XCoordRight](#) [get, set]  
*The normalized x coordinate of the right eye.*
- double [YCoordRight](#) [get, set]  
*The normalized y coordinate of the right eye.*
- double [ZCoordRight](#) [get, set]  
*The normalized z coordinate of the right eye.*

## Events

- PropertyChangedEventHandler? [PropertyChanged](#)  
*The property change event handler.*

### 8.60.1 Detailed Description

The user position to be rendered on the screen.

### 8.60.2 Constructor & Destructor Documentation

#### 8.60.2.1 [UserPositionData\(\)](#) [1/2]

```
GazeUtilityLibrary.DataStructs.UserPositionData.UserPositionData ( ) [inline]
```

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

#### 8.60.2.2 [UserPositionData\(\)](#) [2/2]

```
GazeUtilityLibrary.DataStructs.UserPositionData.UserPositionData (
    double xCoordLeft,
    double yCoordLeft,
    double zCoordLeft,
    double xCoordRight,
    double yCoordRight,
    double zCoordRight ) [inline]
```

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

**Parameters**

<i>xCoordLeft</i>	The normalized x coordinate of the left eye.
<i>yCoordLeft</i>	The normalized y coordinate of the left eye.
<i>zCoordLeft</i>	The normalized z coordinate of the left eye.
<i>xCoordRight</i>	The normalized x coordinate of the right eye.
<i>yCoordRight</i>	The normalized y coordinate of the right eye.
<i>zCoordRight</i>	The normalized z coordinate of the right eye.

### 8.60.3 Property Documentation

#### 8.60.3.1 XCoordLeft

```
double GazeUtilityLibrary.DataStructs.UserPositionData.XCoordLeft [get], [set]
```

The normalized x coordinate of the left eye.

#### 8.60.3.2 XCoordRight

```
double GazeUtilityLibrary.DataStructs.UserPositionData.XCoordRight [get], [set]
```

The normalized x coordinate of the right eye.

#### 8.60.3.3 YCoordLeft

```
double GazeUtilityLibrary.DataStructs.UserPositionData.YCoordLeft [get], [set]
```

The normalized y coordinate of the left eye.

#### 8.60.3.4 YCoordRight

```
double GazeUtilityLibrary.DataStructs.UserPositionData.YCoordRight [get], [set]
```

The normalized y coordinate of the right eye.

### 8.60.3.5 ZCoordLeft

```
double GazeUtilityLibrary.DataStructs.UserPositionData.ZCoordLeft [get], [set]
```

The normalized z coordinate of the left eye.

### 8.60.3.6 ZCoordRight

```
double GazeUtilityLibrary.DataStructs.UserPositionData.ZCoordRight [get], [set]
```

The normalized z coordinate of the right eye.

## 8.60.4 Event Documentation

### 8.60.4.1 PropertyChanged

```
PropertyChangedEventHandler? GazeUtilityLibrary.DataStructs.UserPositionData.PropertyChanged
```

The property change event handler.

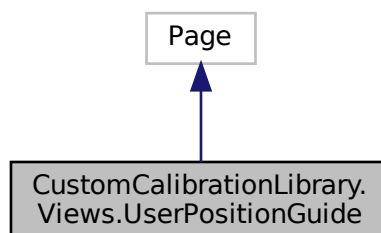
The documentation for this class was generated from the following file:

- source/GazeUtilityLibrary/DataStructs/UserPositionData.cs

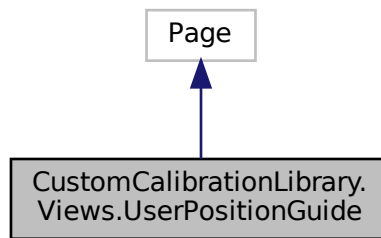
## 8.61 CustomCalibrationLibrary.Views.UserPositionGuide Class Reference

Interaction logic for UserPositionGuide.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.UserPositionGuide:



Collaboration diagram for CustomCalibrationLibrary.Views.UserPositionGuide:



## Public Member Functions

- [UserPositionGuide](#) ([CalibrationModel](#) model)  
*Initializes a new instance of the [UserPositionGuide](#) class.*

### 8.61.1 Detailed Description

Interaction logic for UserPositionGuide.xaml

### 8.61.2 Constructor & Destructor Documentation

#### 8.61.2.1 UserPositionGuide()

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

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

#### Parameters

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

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

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

## 8.62 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*

#### 8.62.1 Detailed Description

The view model class for the user position guide view.

#### 8.62.2 Constructor & Destructor Documentation

##### 8.62.2.1 UserPositionGuideViewModel()

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

Constructor

Parameters

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

##### 8.62.3 Property Documentation

### 8.62.3.1 CalibrationAbortCommand

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

Command to abort the calibration

### 8.62.3.2 CalibrationStartCommand

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

Command to start the calibration

### 8.62.3.3 UserPosition

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

The user position to be represented on the view

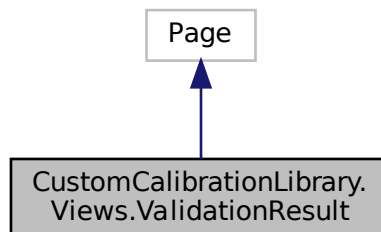
The documentation for this class was generated from the following file:

- source/CustomCalibrationLibrary/ViewModels/UserPositionGuideViewModel.cs

## 8.63 CustomCalibrationLibrary.Views.ValidationResult Class Reference

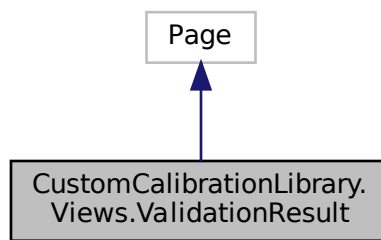
Interaction logic for ValidationResult.xaml

Inheritance diagram for CustomCalibrationLibrary.Views.ValidationResult:





Collaboration diagram for CustomCalibrationLibrary.Views.ValidationResult:



## Public Member Functions

- [ValidationResult](#) ([CalibrationModel](#) model)  
*Initializes a new instance of the [ValidationResult](#) class.*

### 8.63.1 Detailed Description

Interaction logic for ValidationResult.xaml

### 8.63.2 Constructor & Destructor Documentation

#### 8.63.2.1 ValidationResult()

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

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

#### Parameters

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

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

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

## 8.64 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*

#### 8.64.1 Detailed Description

View model class of the gaze validation result.

#### 8.64.2 Constructor & Destructor Documentation

##### 8.64.2.1 ValidationResultViewModel()

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

Constructor

Parameters

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

#### 8.64.3 Property Documentation

### 8.64.3.1 ValidationCloseCommand

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

Command to close the validation window

### 8.64.3.2 ValidationData

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

The validation result

### 8.64.3.3 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

- [\\_model](#)
    - [CustomCalibrationLibrary.ViewModels.CalibrationViewModel, 89](#)
- [AccuracyLeft](#)
  - [GazeUtilityLibrary.DataStructs.GazeValidationData, 145](#)
- [AccuracyLeftEye](#)
  - [Tobii.Research.Addons.CalibrationValidationPoint, 83](#)
- [AccuracyRight](#)
  - [GazeUtilityLibrary.DataStructs.GazeValidationData, 146](#)
- [AccuracyRightEye](#)
  - [Tobii.Research.Addons.CalibrationValidationPoint, 83](#)
- [AddPoint](#)
  - [GazeUtilityLibrary.DataStructs.GazeValidationData, 145](#)
- [App](#)
  - [GazeToMouse.App, 33](#)
- [ApplyCalibration](#)
  - [GazeUtilityLibrary.Tracker.BaseTracker, 42](#)
  - [GazeUtilityLibrary.Tracker.EyeTrackerPro, 114](#)
  - [GazeUtilityLibrary.Tracker.MouseTracker, 160](#)
- [AverageAccuracyLeftEye](#)
  - [Tobii.Research.Addons.CalibrationValidationResult, 86](#)
- [AverageAccuracyRightEye](#)
  - [Tobii.Research.Addons.CalibrationValidationResult, 86](#)
- [AveragePrecisionLeftEye](#)
  - [Tobii.Research.Addons.CalibrationValidationResult, 86](#)
- [AveragePrecisionRightEye](#)
  - [Tobii.Research.Addons.CalibrationValidationResult, 86](#)
- [AveragePrecisionRMSLeftEye](#)
  - [Tobii.Research.Addons.CalibrationValidationResult, 86](#)
- [AveragePrecisionRMSRightEye](#)
  - [Tobii.Research.Addons.CalibrationValidationResult, 86](#)
- [BaseTracker](#)
  - [GazeUtilityLibrary.Tracker.BaseTracker, 41](#)
- [BottomLeft](#)
  - [GazeUtilityLibrary.ConfigScreenArea, 102](#)
  - [GazeUtilityLibrary.ScreenArea, 175](#)
- [BottomRight](#)
  - [GazeUtilityLibrary.ConfigScreenArea, 102](#)
  - [GazeUtilityLibrary.ScreenArea, 175](#)
- [Calibration](#)
  - [CustomCalibrationLibrary.Views.Calibration, 54](#)
- [CalibrationAbortCommand](#)
  - [CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel, 183](#)
- [CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel, 195](#)
- [CustomCalibrationLibrary.Views.CalibrationFailed, 60](#)
- [CustomCalibrationLibrary.Views.Disconnect, 104](#)
- [CalibrationAcceptCommand](#)
  - [CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, 81](#)
- [CalibrationCommand](#)
  - [CustomCalibrationLibrary.Commands.CalibrationCommand, 56](#)
- [CalibrationEvent](#)
  - [CustomCalibrationLibrary.Models.CalibrationModel, 69](#)
- [CalibrationEventType](#)
  - [CustomCalibrationLibrary.Models, 24](#)
- [CalibrationFailed](#)
  - [CustomCalibrationLibrary.Views.CalibrationFailed, 60](#)
- [CalibrationFrame](#)
  - [CustomCalibrationLibrary.Views.CalibrationFrame, 62](#)
- [CalibrationLogColumnOrder](#)
  - [GazeUtilityLibrary.ConfigItem, 94](#)
- [CalibrationLogColumnName](#)
  - [GazeUtilityLibrary.ConfigItem, 94](#)
- [CalibrationLogWriteOutput](#)
  - [GazeUtilityLibrary.ConfigItem, 94](#)
- [CalibrationModel](#)
  - [CustomCalibrationLibrary.Models.CalibrationModel, 65](#)
- [CalibrationOutputValue](#)
  - [GazeUtilityLibrary.DataStructs, 29](#)
- [CalibrationPoint](#)
  - [CustomCalibrationLibrary.Views.CalibrationPoint, 74](#)
  - [GazeUtilityLibrary.DataStructs.CalibrationPoint, 71](#)
- [CalibrationPoints](#)
  - [CustomCalibrationLibrary.Models.CalibrationModel, 67](#)
  - [CustomCalibrationLibrary.ViewModels.CalibrationViewModel, 89](#)

- GazeUtilityLibrary.ConfigItem, 94
- CalibrationPointViewModel
  - CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel, 75, 76
- CalibrationRestartCommand
  - CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, 81
  - CustomCalibrationLibrary.Views.CalibrationFailed, 61
- CalibrationResult
  - CustomCalibrationLibrary.Views.CalibrationResult, 77
- CalibrationResultPoint
  - CustomCalibrationLibrary.Views.CalibrationResultPoint, 78
- CalibrationResultViewModel
  - CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, 81
- CalibrationStartCommand
  - CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel, 183
  - CustomCalibrationLibrary.ViewModels.UserPositionGuidanceViewModel, 196
- CalibrationStatus
  - CustomCalibrationLibrary.Models, 24
- CalibrationValidate
  - GazeToMouse.App, 34
- CalibrationViewModel
  - CustomCalibrationLibrary.ViewModels.CalibrationViewModel, 88
- CanExecute
  - CustomCalibrationLibrary.Commands.CalibrationCommand, 56
- CanExecuteChanged
  - CustomCalibrationLibrary.Commands.CalibrationCommand, 57
- Center
  - GazeUtilityLibrary.ConfigScreenArea, 102
  - GazeUtilityLibrary.ScreenArea, 175
- CleanupCalibrationOutputFile
  - GazeUtilityLibrary.GazeConfiguration, 127
- CleanupGazeOutputFile
  - GazeUtilityLibrary.GazeConfiguration, 127
- CleanupValidationOutputFile
  - GazeUtilityLibrary.GazeConfiguration, 127
- CollectCalibrationDataAsync
  - GazeUtilityLibrary.Tracker.BaseTracker, 42
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 115
  - GazeUtilityLibrary.Tracker.MouseTracker, 160
- CollectValidationDataAsync
  - GazeUtilityLibrary.Tracker.BaseTracker, 42
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 115
  - GazeUtilityLibrary.Tracker.MouseTracker, 160
- Combined
  - GazeUtilityLibrary.DataStructs.GazeData, 134
- Command
  - GazeUtilityLibrary.DataStructs.PipeCommand, 165
- CompensateDrift
  - GazeToMouse.App, 34
- Compensation
  - GazeUtilityLibrary.DataStructs.DriftCompensationData, 108
- Compute
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 179
- ComputeValidation
  - GazeUtilityLibrary.Tracker.BaseTracker, 44
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 115
  - GazeUtilityLibrary.Tracker.MouseTracker, 161
- Computing
  - CustomCalibrationLibrary.Views.Computing, 91
- Config
  - GazeUtilityLibrary.GazeConfiguration, 130
- config
  - GazeUtilityLibrary.Tracker.BaseTracker, 50
- ConfigItem
  - GazeUtilityLibrary.ConfigItem, 93
  - GazeUtilityLibrary.ConfigItem, 94
  - GazeUtilityLibrary.ConfigScreenArea, 101
- Convert
  - CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter, 150
  - CustomCalibrationLibrary.Converters.PositionConverter, 167
  - CustomCalibrationLibrary.Converters.ProximityColorConverter, 170
- ConvertBack
  - CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter, 150
  - CustomCalibrationLibrary.Converters.PositionConverter, 167
  - CustomCalibrationLibrary.Converters.ProximityColorConverter, 170
- ConvertToBinString
  - GazeUtilityLibrary.GazeError, 143
- Coordinates
  - Tobii.Research.Addons.CalibrationValidationPoint, 83
- CustomCalibrate
  - GazeToMouse.App, 34
- CustomCalibrationLibrary, 23
- CustomCalibrationLibrary.Commands, 23
- CustomCalibrationLibrary.Commands.CalibrationCommand, 55
  - CalibrationCommand, 56
  - CanExecute, 56
  - CanExecuteChanged, 57
  - Execute, 56
- CustomCalibrationLibrary.Converters, 23
- CustomCalibrationLibrary.Converters.HasDataToVisibilityConverter, 149
  - Convert, 150
  - ConvertBack, 150
- CustomCalibrationLibrary.Converters.PositionConverter,

- 166
- Convert, 167
- ConvertBack, 167
- Offset, 168
- OffsetProperty, 168
- CustomCalibrationLibrary.Converters.ProximityColorConverter, 169
- Convert, 170
- ConvertBack, 170
- CustomCalibrationLibrary.Models, 23
  - CalibrationEventType, 24
  - CalibrationStatus, 24
- CustomCalibrationLibrary.Models.CalibrationModel, 63
  - CalibrationEvent, 69
  - CalibrationModel, 65
  - CalibrationPoints, 67
  - Error, 67
  - GazeDataCollected, 65
  - GazePoint, 67
  - GazePointChanged, 69
  - Index, 67
  - InitCalibration, 66
  - LastStatus, 68
  - NextCalibrationPoint, 66
  - OnCalibrationEvent, 66
  - Points, 68
  - PropertyChanged, 69
  - RedoCalibrationPoint, 66
  - SetCalibrationResult, 66
  - Status, 68
  - UpdateGazePoint, 67
  - UserPositionGuide, 68
  - UserPositionGuideChanged, 69
  - ValidationData, 68
- CustomCalibrationLibrary.ViewModels, 24
- CustomCalibrationLibrary.ViewModels.CalibrationPointViewModel, 74
  - CalibrationPointViewModel, 75, 76
- CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, 79
  - CalibrationAcceptCommand, 81
  - CalibrationRestartCommand, 81
  - CalibrationResultViewModel, 81
  - GazePoint, 81
  - GazeVisibilityCommand, 82
  - OnGazeToggle, 81
- CustomCalibrationLibrary.ViewModels.CalibrationViewModel, 87
  - \_model, 89
  - CalibrationPoints, 89
  - CalibrationViewModel, 88
- CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel, 108
  - DriftCompensationViewModel, 109
  - FixationPoint, 109
- CustomCalibrationLibrary.ViewModels.Monitor, 155
  - Index, 155
  - Monitor, 155
  - Name, 156
- CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel, 182
  - CalibrationAbortCommand, 183
  - CalibrationStartCommand, 183
  - Monitors, 183
  - ScreenSelectionViewModel, 183
  - ScreenSwitchCommand, 184
- CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel, 195
  - CalibrationAbortCommand, 195
  - CalibrationStartCommand, 196
  - UserPosition, 196
  - UserPositionGuideViewModel, 195
- CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, 198
  - ValidationCloseCommand, 198
  - ValidationData, 199
  - ValidationRestartCommand, 199
  - ValidationResultViewModel, 198
- CustomCalibrationLibrary.Views, 25
- CustomCalibrationLibrary.Views.Calibration, 53
  - Calibration, 54
- CustomCalibrationLibrary.Views.CalibrationFailed, 59
  - CalibrationAbortCommand, 60
  - CalibrationFailed, 60
  - CalibrationRestartCommand, 61
  - Error, 61
  - PropertyChanged, 61
- CustomCalibrationLibrary.Views.CalibrationFrame, 62
  - CalibrationFrame, 62
- CustomCalibrationLibrary.Views.CalibrationPoint, 73
  - CalibrationPoint, 74
- CustomCalibrationLibrary.Views.CalibrationResult, 76
  - CalibrationResult, 77
- CustomCalibrationLibrary.Views.CalibrationResultPoint, 78
  - CalibrationResultPoint, 78
- CustomCalibrationLibrary.Views.CalibrationWindow, 89
- CustomCalibrationLibrary.Views.Computing, 90
  - Computing, 91
- CustomCalibrationLibrary.Views.Disconnect, 103
  - CalibrationAbortCommand, 104
  - Disconnect, 104
- CustomCalibrationLibrary.Views.DriftCompensationWindow, 110
  - DriftCompensationWindow, 110
- CustomCalibrationLibrary.Views.FixationPoint, 119
  - FixationPoint, 120
- CustomCalibrationLibrary.Views.ScreenSelection, 181
  - ScreenSelection, 182
- CustomCalibrationLibrary.Views.UserPositionGuide, 193
  - UserPositionGuide, 194
- CustomCalibrationLibrary.Views.ValidationResult, 196
  - ValidationResult, 197
- DataLogColumnOrder
  - GazeUtilityLibrary.ConfigItem, 94

- DataLogColumnNameTitle
  - GazeUtilityLibrary.ConfigItem, 95
- DataLogCount
  - GazeUtilityLibrary.ConfigItem, 95
- DataLogDisabledOnStartup
  - GazeUtilityLibrary.ConfigItem, 95
- DataLogFormatDiameter
  - GazeUtilityLibrary.ConfigItem, 95
- DataLogFormatNormalizedPoint
  - GazeUtilityLibrary.ConfigItem, 95
- DataLogFormatOrigin
  - GazeUtilityLibrary.ConfigItem, 95
- DataLogFormatTimeStamp
  - GazeUtilityLibrary.ConfigItem, 96
- DataLogFormatTimeStampRelative
  - GazeUtilityLibrary.ConfigItem, 96
- DataLogFormatValidation
  - GazeUtilityLibrary.ConfigItem, 96
- DataLogPath
  - GazeUtilityLibrary.ConfigItem, 96
- DataLogWriteOutput
  - GazeUtilityLibrary.ConfigItem, 96
- Debug
  - GazeUtilityLibrary.TrackerLogger, 187
- DeviationAngle
  - GazeUtilityLibrary.DriftCompensation, 106
- DeviceName
  - GazeUtilityLibrary.Tracker.BaseTracker, 51
- DeviceStatus
  - GazeUtilityLibrary.Tracker.BaseTracker, 41
- dialogBoxTimer
  - GazeUtilityLibrary.Tracker.BaseTracker, 51
- DiscardData
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 179
- Disconnect
  - CustomCalibrationLibrary.Views.Disconnect, 104
- Dispersion
  - GazeUtilityLibrary.DriftCompensation, 106
- Dispose
  - GazeUtilityLibrary.Tracker.BaseTracker, 44
  - GazeUtilityLibrary.Tracker.MouseTracker, 161
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 179
- DriftCompensation
  - GazeUtilityLibrary.DataStructs.GazeData, 134
  - GazeUtilityLibrary.DriftCompensation, 105
- driftCompensation
  - GazeUtilityLibrary.Tracker.BaseTracker, 51
- DriftCompensationComputed
  - GazeUtilityLibrary.Tracker.BaseTracker, 52
- DriftCompensationData
  - GazeUtilityLibrary.DataStructs.DriftCompensationData, 107
- DriftCompensationDispersionThreshold
  - GazeUtilityLibrary.ConfigItem, 96
- DriftCompensationDispersionThresholdMax
  - GazeUtilityLibrary.ConfigItem, 97
- DriftCompensationDurationThreshold
  - GazeUtilityLibrary.ConfigItem, 97
- DriftCompensationEventHandler
  - GazeUtilityLibrary.Tracker.BaseTracker, 45
- DriftCompensationTimer
  - GazeUtilityLibrary.ConfigItem, 97
- DriftCompensationViewModel
  - CustomCalibrationLibrary.ViewModels.DriftCompensationViewModel, 109
- DriftCompensationWindow
  - CustomCalibrationLibrary.Views.DriftCompensationWindow, 110
- DriftCompensationWindowShow
  - GazeUtilityLibrary.ConfigItem, 97
- Dump
  - GazeUtilityLibrary.ScreenArea, 172
- DumpCurrentConfigurationFile
  - GazeUtilityLibrary.GazeConfiguration, 128
- DumpFatal
  - GazeUtilityLibrary.TrackerLogger, 188
- E1
  - GazeUtilityLibrary.ScreenTriangle, 186
- E2
  - GazeUtilityLibrary.ScreenTriangle, 186
- ECalibrationDataError
  - GazeUtilityLibrary, 27
- EGazeConfigError
  - GazeUtilityLibrary, 27
- EGazeDataError
  - GazeUtilityLibrary, 27
- EnterValidationMode
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 179
- EOutputType
  - GazeUtilityLibrary, 27
- Error
  - CustomCalibrationLibrary.Models.CalibrationModel, 67
  - CustomCalibrationLibrary.Views.CalibrationFailed, 61
  - GazeUtilityLibrary.CalibrationDataError, 59
  - GazeUtilityLibrary.GazeConfigError, 125
  - GazeUtilityLibrary.GazeDataError, 142
  - GazeUtilityLibrary.TrackerLogger, 188
- Execute
  - CustomCalibrationLibrary.Commands.CalibrationCommand, 56
- EyeData
  - GazeUtilityLibrary.DataStructs.EyeData, 111
  - GazeUtilityLibrary.DataStructs.GazeDataCollection, 140
- EyeTrackerPro
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 114
- FinishCalibration
  - GazeUtilityLibrary.Tracker.BaseTracker, 45
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 116
  - GazeUtilityLibrary.Tracker.MouseTracker, 161



- FinishCalibrationAsync
  - GazeUtilityLibrary.Tracker.BaseTracker, [45](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [116](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [161](#)
- FinishValidation
  - GazeUtilityLibrary.Tracker.BaseTracker, [45](#)
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, [116](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [162](#)
- FixationPoint
  - CustomCalibrationLibrary.ViewModels.DriftCompensationView, [109](#)
  - CustomCalibrationLibrary.Views.FixationPoint, [120](#)
- GazeCalibrationData
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [121](#)
- GazeConfiguration
  - GazeUtilityLibrary.GazeConfiguration, [126](#)
- GazeControl, [25](#)
- GazeControl.App, [31](#)
- GazeData
  - GazeUtilityLibrary.DataStructs.GazeData, [131](#), [132](#)
  - Tobii.Research.Addons.CalibrationValidationPoint, [83](#)
- GazeData2d
  - GazeUtilityLibrary.DataStructs.GazeData2d, [135](#)
  - GazeUtilityLibrary.DataStructs.GazeDataCollection, [140](#)
- GazeData3d
  - GazeUtilityLibrary.DataStructs.GazeData3d, [137](#)
  - GazeUtilityLibrary.DataStructs.GazeDataCollection, [140](#)
- GazeDataCollected
  - CustomCalibrationLibrary.Models.CalibrationModel, [65](#)
- GazeDataCollection
  - GazeUtilityLibrary.DataStructs.GazeDataCollection, [139](#)
- GazeDataHandler
  - GazeUtilityLibrary.Tracker.BaseTracker, [45](#)
- GazeDataReceived
  - GazeUtilityLibrary.Tracker.BaseTracker, [52](#)
- GazeDirection
  - GazeUtilityLibrary.DataStructs.GazeData3d, [137](#)
- GazeDistance
  - GazeUtilityLibrary.DataStructs.GazeData3d, [137](#)
- GazeOrigin
  - GazeUtilityLibrary.DataStructs.GazeData3d, [137](#)
- GazeOutputValue
  - GazeUtilityLibrary.DataStructs, [29](#)
- GazePoint
  - CustomCalibrationLibrary.Models.CalibrationModel, [67](#)
  - CustomCalibrationLibrary.ViewModels.CalibrationResultView, [81](#)
  - GazeUtilityLibrary.DataStructs.GazeData2d, [136](#)
  - GazeUtilityLibrary.DataStructs.GazeData3d, [138](#)
- GazePointChanged
  - CustomCalibrationLibrary.Models.CalibrationModel, [69](#)
- GazePosition2d
  - GazeUtilityLibrary.DataStructs.DriftCompensationData, [108](#)
- GazePosition3d
  - GazeUtilityLibrary.DataStructs.DriftCompensationData, [108](#)
- GazePositionAverage
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, [71](#)
- GazePositionLeft
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, [72](#)
- GazePositionRight
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, [72](#)
- GazeRecordingDisable
  - GazeToMouse.App, [34](#)
- GazeRecordingEnable
  - GazeToMouse.App, [34](#)
- GazeToMouse, [25](#)
- GazeToMouse.App, [32](#)
  - App, [33](#)
  - CalibrationValidate, [34](#)
  - CompensateDrift, [34](#)
  - CustomCalibrate, [34](#)
  - GazeRecordingDisable, [34](#)
  - GazeRecordingEnable, [34](#)
  - LastTag, [35](#)
  - MouseTrackingDisable, [35](#)
  - MouseTrackingEnable, [35](#)
  - ResetDriftCompensation, [35](#)
  - StartTime, [35](#)
  - Tag, [35](#)
  - TrialId, [36](#)
- GazeUtilityLibrary, [26](#)
  - ECalibrationDataError, [27](#)
  - EGazeConfigError, [27](#)
  - EGazeDataError, [27](#)
  - EOutputType, [27](#)
- GazeUtilityLibrary.CalibrationDataError, [57](#)
  - Error, [59](#)
- GazeUtilityLibrary.GetCalibrationDataErrorString, [58](#)
- GazeUtilityLibrary.ConfigItem, [91](#)
  - CalibrationLogColumnOrder, [94](#)
  - CalibrationLogColumnTitle, [94](#)
  - CalibrationLogWriteOutput, [94](#)
  - CalibrationPoints, [94](#)
  - ConfigItem, [93](#)
  - ConfigName, [94](#)
  - DataLogColumnOrder, [94](#)
  - DataLogColumnTitle, [95](#)
  - DataLogCount, [95](#)
  - DataLogDisabledOnStartup, [95](#)
  - DataLogFormatDiameter, [95](#)
  - DataLogFormatNormalizedPoint, [95](#)
  - DataLogFormatOrigin, [95](#)
  - DataLogFormatTimeStamp, [96](#)
  - DataLogFormatTimeStampRelative, [96](#)
  - DataLogFormatValidation, [96](#)

- DataLogPath, 96
- DataLogWriteOutput, 96
- DriftCompensationDispersionThreshold, 96
- DriftCompensationDispersionThresholdMax, 97
- DriftCompensationDurationThreshold, 97
- DriftCompensationTimer, 97
- DriftCompensationWindowShow, 97
- LicensePath, 97
- MouseCalibrationHide, 97
- MouseControl, 98
- MouseControlHide, 98
- MouseStandardIconPath, 98
- ReadyTimer, 98
- ScreenArea, 98
- TobiiApplicationPath, 98
- TobiiCalibrate, 99
- TobiiCalibrateArguments, 99
- TrackerDevice, 99
- ValidationDurationThreshold, 99
- ValidationLogColumnOrder, 99
- ValidationLogColumnTitle, 99
- ValidationLogWriteOutput, 100
- ValidationPoints, 100
- ValidationTimer, 100
- GazeUtilityLibrary.ConfigScreenArea, 100
  - BottomLeft, 102
  - BottomRight, 102
  - Center, 102
  - ConfigScreenArea, 101
  - Height, 102
  - TopLeft, 102
  - TopRight, 102
  - Width, 103
- GazeUtilityLibrary.DataStructs, 28
  - CalibrationOutputValue, 29
  - GazeOutputValue, 29
  - ValidationOutputValue, 29
- GazeUtilityLibrary.DataStructs.CalibrationPoint, 70
  - CalibrationPoint, 71
  - GazePositionAverage, 71
  - GazePositionLeft, 72
  - GazePositionRight, 72
  - HasData, 72
  - Index, 72
  - Position, 72
  - PropertyChanged, 73
- GazeUtilityLibrary.DataStructs.DriftCompensationData, 107
  - Compensation, 108
  - DriftCompensationData, 107
  - GazePosition2d, 108
  - GazePosition3d, 108
- GazeUtilityLibrary.DataStructs.EyeData, 111
  - EyeData, 111
  - IsPupilDiameterValid, 112
  - PupilDiameter, 112
- GazeUtilityLibrary.DataStructs.GazeCalibrationData, 120
  - GazeCalibrationData, 121
  - Prepare, 122
  - ValidityLeft, 122
  - ValidityRight, 122
  - XCoord, 122
  - XCoordLeft, 122
  - XCoordRight, 123
  - YCoord, 123
  - YCoordLeft, 123
  - YCoordRight, 123
- GazeUtilityLibrary.DataStructs.GazeData, 131
  - Combined, 134
  - DriftCompensation, 134
  - GazeData, 131, 132
  - Left, 134
  - Prepare, 133
  - Right, 134
  - Timestamp, 134
  - TimestampReceived, 134
- GazeUtilityLibrary.DataStructs.GazeData2d, 135
  - GazeData2d, 135
  - GazePoint, 136
  - IsGazePointValid, 136
- GazeUtilityLibrary.DataStructs.GazeData3d, 136
  - GazeData3d, 137
  - GazeDirection, 137
  - GazeDistance, 137
  - GazeOrigin, 137
  - GazePoint, 138
  - IsGazeOriginValid, 138
  - IsGazePointValid, 138
- GazeUtilityLibrary.DataStructs.GazeDataCollection, 138
  - EyeData, 140
  - GazeData2d, 140
  - GazeData3d, 140
  - GazeDataCollection, 139
- GazeUtilityLibrary.DataStructs.GazeValidationData, 143
  - AccuracyLeft, 145
  - AccuracyRight, 146
  - AddPoint, 145
  - GazeValidationData, 144
  - Points, 146
  - PrecisionLeft, 146
  - PrecisionRight, 146
  - PrecisionRmsLeft, 146
  - PrecisionRmsRight, 146
- GazeUtilityLibrary.DataStructs.GazeValidationPoint, 147
  - GazeValidationPoint, 147
  - Point, 148
  - Prepare, 148
  - Result, 148
- GazeUtilityLibrary.DataStructs.LiveGazePoint, 153
  - PropertyChanged, 154
  - Visibility, 154
  - X, 154
  - Y, 154
- GazeUtilityLibrary.DataStructs.PipeCommand, 164
  - Command, 165

- Label, [165](#)
- PipeCommand, [164](#)
- ResetStartTime, [165](#)
- TrialId, [165](#)
- GazeUtilityLibrary.DataStructs.UserPositionData, [190](#)
  - PropertyChanged, [193](#)
  - UserPositionData, [191](#)
  - XCoordLeft, [192](#)
  - XCoordRight, [192](#)
  - YCoordLeft, [192](#)
  - YCoordRight, [192](#)
  - ZCoordLeft, [192](#)
  - ZCoordRight, [193](#)
- GazeUtilityLibrary.DriftCompensation, [105](#)
  - DeviationAngle, [106](#)
  - Dispersion, [106](#)
  - DriftCompensation, [105](#)
  - Q, [107](#)
  - Reset, [106](#)
  - Start, [106](#)
  - Update, [106](#)
- GazeUtilityLibrary.GazeConfigError, [124](#)
  - Error, [125](#)
  - GetGazeConfigErrorString, [125](#)
- GazeUtilityLibrary.GazeConfiguration, [125](#)
  - CleanupCalibrationOutputFile, [127](#)
  - CleanupGazeOutputFile, [127](#)
  - CleanupValidationOutputFile, [127](#)
  - Config, [130](#)
  - DumpCurrentConfigurationFile, [128](#)
  - GazeConfiguration, [126](#)
  - InitConfig, [128](#)
  - PrepareCalibrationOutputFile, [128](#)
  - PrepareGazeOutputFile, [129](#)
  - PrepareValidationOutputFile, [129](#)
  - WriteToCalibrationOutput, [129](#)
  - WriteToGazeOutput, [130](#)
  - WriteToValidationOutput, [130](#)
- GazeUtilityLibrary.GazeDataError, [141](#)
  - Error, [142](#)
  - GetGazeDataErrorString, [142](#)
- GazeUtilityLibrary.GazeError, [142](#)
  - ConvertToBinString, [143](#)
- GazeUtilityLibrary.JsonConfigParser, [151](#)
  - GetDefaultConfig, [152](#)
  - JsonConfigParser, [151](#)
  - ParseJsonConfig, [152](#)
  - SerializeJsonConfig, [152](#)
- GazeUtilityLibrary.MouseHider, [156](#)
  - HideCursor, [157](#)
  - MouseHider, [156](#)
  - ShowCursor, [157](#)
- GazeUtilityLibrary.ScreenArea, [171](#)
  - BottomLeft, [175](#)
  - BottomRight, [175](#)
  - Center, [175](#)
  - Dump, [172](#)
  - GetIntersectionPoint, [173](#)
  - GetPoint2d, [173](#)
  - GetPoint2dNormalized, [173](#)
  - Height, [175](#)
  - ScreenArea, [172](#)
  - TopLeft, [175](#)
  - TopRight, [176](#)
  - Width, [176](#)
- GazeUtilityLibrary.ScreenTriangle, [184](#)
  - E1, [186](#)
  - E2, [186](#)
  - GetIntersectionPoint, [185](#)
  - ScreenTriangle, [185](#)
  - V1, [186](#)
  - V2, [186](#)
  - V3, [186](#)
- GazeUtilityLibrary.Tracker, [29](#)
- GazeUtilityLibrary.Tracker.BaseTracker, [38](#)
  - ApplyCalibration, [42](#)
  - BaseTracker, [41](#)
  - CollectCalibrationDataAsync, [42](#)
  - CollectValidationDataAsync, [42](#)
  - ComputeValidation, [44](#)
  - config, [50](#)
  - DeviceName, [51](#)
  - DeviceStatus, [41](#)
  - dialogBoxTimer, [51](#)
  - Dispose, [44](#)
  - driftCompensation, [51](#)
  - DriftCompensationComputed, [52](#)
  - DriftCompensationEventHandler, [45](#)
  - FinishCalibration, [45](#)
  - FinishCalibrationAsync, [45](#)
  - FinishValidation, [45](#)
  - GazeDataHandler, [45](#)
  - GazeDataReceived, [52](#)
  - GetFixationFrameCount, [46](#)
  - GetUnitDirection, [46](#)
  - InitCalibration, [46](#)
  - InitCalibrationAsync, [46](#)
  - InitDriftCompensation, [47](#)
  - InitValidation, [47](#)
  - IsInitialised, [47](#)
  - IsReady, [47](#)
  - logger, [51](#)
  - OnGazeDataReceived, [48](#)
  - OnPropertyChanged, [48](#)
  - OnTrackerDisabled, [48](#)
  - OnTrackerDisabledTimeout, [49](#)
  - OnTrackerEnabled, [49](#)
  - OnUserPositionDataReceived, [49](#)
  - PatternReplace, [49](#)
  - PropertyChanged, [52](#)
  - ResetDriftCompensation, [50](#)
  - ScreenArea, [52](#)
  - screenArea, [51](#)
  - StartDriftCompensation, [50](#)
  - State, [52](#)
  - TrackerDisabled, [52](#)

- TrackerEnabled, 53
- trackerMessageBox, 51
- UserPositionDataHandler, 50
- UserPositionDataReceived, 53
- GazeUtilityLibrary.Tracker.EyeTrackerPro, 112
  - ApplyCalibration, 114
  - CollectCalibrationDataAsync, 115
  - CollectValidationDataAsync, 115
  - ComputeValidation, 115
  - EyeTrackerPro, 114
  - FinishCalibration, 116
  - FinishCalibrationAsync, 116
  - FinishValidation, 116
  - GetFixationFrameCount, 116
  - GetUnitDirection, 117
  - InitCalibration, 117
  - InitCalibrationAsync, 117
  - InitDriftCompensation, 117
  - InitValidation, 118
  - IsInitialised, 118
  - IsLicenseOk, 118
  - PatternReplace, 118
- GazeUtilityLibrary.Tracker.MouseTracker, 157
  - ApplyCalibration, 160
  - CollectCalibrationDataAsync, 160
  - CollectValidationDataAsync, 160
  - ComputeValidation, 161
  - Dispose, 161
  - FinishCalibration, 161
  - FinishCalibrationAsync, 161
  - FinishValidation, 162
  - GetFixationFrameCount, 162
  - GetUnitDirection, 162
  - InitCalibration, 163
  - InitCalibrationAsync, 163
  - InitDriftCompensation, 163
  - InitValidation, 163
  - MouseTracker, 159
  - Start, 163
  - Stop, 164
- GazeUtilityLibrary.TrackerLogger, 187
  - Debug, 187
  - DumpFatal, 188
  - Error, 188
  - Info, 188
  - TrackerLogger, 187
  - Warning, 188
- GazeUtilityLibrary.TrackerMessageBox, 189
- GazeValidationData
  - GazeUtilityLibrary.DataStructs.GazeValidationData, 144
- GazeValidationPoint
  - GazeUtilityLibrary.DataStructs.GazeValidationPoint, 147
- GazeVisibilityCommand
  - CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, 82
- GetCalibrationDataErrorString
  - GazeUtilityLibrary.CalibrationDataError, 58
- GetDefaultConfig
  - GazeUtilityLibrary.JsonConfigParser, 152
- GetFixationFrameCount
  - GazeUtilityLibrary.Tracker.BaseTracker, 46
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 116
  - GazeUtilityLibrary.Tracker.MouseTracker, 162
- GetGazeConfigErrorString
  - GazeUtilityLibrary.GazeConfigError, 125
- GetGazeDataErrorString
  - GazeUtilityLibrary.GazeDataError, 142
- GetIntersectionPoint
  - GazeUtilityLibrary.ScreenArea, 173
  - GazeUtilityLibrary.ScreenTriangle, 185
- GetPoint2d
  - GazeUtilityLibrary.ScreenArea, 173
- GetPoint2dNormalized
  - GazeUtilityLibrary.ScreenArea, 173
- GetUnitDirection
  - GazeUtilityLibrary.Tracker.BaseTracker, 46
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 117
  - GazeUtilityLibrary.Tracker.MouseTracker, 162
- HasData
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, 72
- Height
  - GazeUtilityLibrary.ConfigScreenArea, 102
  - GazeUtilityLibrary.ScreenArea, 175
- HideCursor
  - GazeUtilityLibrary.MouseHider, 157
- Index
  - CustomCalibrationLibrary.Models.CalibrationModel, 67
  - CustomCalibrationLibrary.ViewModels.Monitor, 155
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, 72
- Info
  - GazeUtilityLibrary.TrackerLogger, 188
- InitCalibration
  - CustomCalibrationLibrary.Models.CalibrationModel, 66
  - GazeUtilityLibrary.Tracker.BaseTracker, 46
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 117
  - GazeUtilityLibrary.Tracker.MouseTracker, 163
- InitCalibrationAsync
  - GazeUtilityLibrary.Tracker.BaseTracker, 46
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 117
  - GazeUtilityLibrary.Tracker.MouseTracker, 163
- InitConfig
  - GazeUtilityLibrary.GazeConfiguration, 128
- InitDriftCompensation
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 117
  - GazeUtilityLibrary.Tracker.MouseTracker, 163
- InitValidation
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 118
  - GazeUtilityLibrary.Tracker.MouseTracker, 163

- IsGazeOriginValid
  - GazeUtilityLibrary.DataStructs.GazeData3d, 138
- IsGazePointValid
  - GazeUtilityLibrary.DataStructs.GazeData2d, 136
  - GazeUtilityLibrary.DataStructs.GazeData3d, 138
- IsInitialised
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 118
- IsLicenseOk
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 118
- IsPupilDiameterValid
  - GazeUtilityLibrary.DataStructs.EyeData, 112
- IsReady
  - GazeUtilityLibrary.Tracker.BaseTracker, 47
- JsonConfigParser
  - GazeUtilityLibrary.JsonConfigParser, 151
- Label
  - GazeUtilityLibrary.DataStructs.PipeCommand, 165
- LastStatus
  - CustomCalibrationLibrary.Models.CalibrationModel, 68
- LastTag
  - GazeToMouse.App, 35
- LeaveValidationMode
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 179
- Left
  - GazeUtilityLibrary.DataStructs.GazeData, 134
- LicensePath
  - GazeUtilityLibrary.ConfigItem, 97
- logger
  - GazeUtilityLibrary.Tracker.BaseTracker, 51
- Monitor
  - CustomCalibrationLibrary.ViewModels.Monitor, 155
- Monitors
  - CustomCalibrationLibrary.ViewModels.ScreenSelectionView, 183
- MouseCalibrationHide
  - GazeUtilityLibrary.ConfigItem, 97
- MouseControl
  - GazeUtilityLibrary.ConfigItem, 98
- MouseControlHide
  - GazeUtilityLibrary.ConfigItem, 98
- MouseHider
  - GazeUtilityLibrary.MouseHider, 156
- MouseStandardIconPath
  - GazeUtilityLibrary.ConfigItem, 98
- MouseTracker
  - GazeUtilityLibrary.Tracker.MouseTracker, 159
- MouseTrackingDisable
  - GazeToMouse.App, 35
- MouseTrackingEnable
  - GazeToMouse.App, 35
- Name
  - CustomCalibrationLibrary.ViewModels.Monitor, 156
- NextCalibrationPoint
  - CustomCalibrationLibrary.Models.CalibrationModel, 66
- Offset
  - CustomCalibrationLibrary.Converters.PositionConverter, 168
- OffsetProperty
  - CustomCalibrationLibrary.Converters.PositionConverter, 168
- OnCalibrationEvent
  - CustomCalibrationLibrary.Models.CalibrationModel, 66
- OnGazeDataReceived
  - GazeUtilityLibrary.Tracker.BaseTracker, 48
- OnGazeToggle
  - CustomCalibrationLibrary.ViewModels.CalibrationResultViewModel, 81
- OnPropertyChanged
  - GazeUtilityLibrary.Tracker.BaseTracker, 48
- OnTrackerDisabled
  - GazeUtilityLibrary.Tracker.BaseTracker, 48
- OnTrackerDisabledTimeout
  - GazeUtilityLibrary.Tracker.BaseTracker, 49
- OnTrackerEnabled
  - GazeUtilityLibrary.Tracker.BaseTracker, 49
- OnUserPositionDataReceived
  - GazeUtilityLibrary.Tracker.BaseTracker, 49
- ParseJsonConfig
  - GazeUtilityLibrary.JsonConfigParser, 152
- PatternReplace
  - GazeUtilityLibrary.Tracker.BaseTracker, 49
  - GazeUtilityLibrary.Tracker.EyeTrackerPro, 118
- PipeCommand
  - GazeUtilityLibrary.DataStructs.PipeCommand, 164
- Point
  - GazeUtilityLibrary.DataStructs.GazeValidationPoint, 148
- Points
  - CustomCalibrationLibrary.Models.CalibrationModel, 68
  - GazeUtilityLibrary.DataStructs.GazeValidationData, 146
  - Tobii.Research.Addons.CalibrationValidationResult, 87
- Position
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, 72
- PrecisionLeft
  - GazeUtilityLibrary.DataStructs.GazeValidationData, 146
- PrecisionLeftEye
  - Tobii.Research.Addons.CalibrationValidationPoint, 84
- PrecisionRight
  - GazeUtilityLibrary.DataStructs.GazeValidationData, 146

- PrecisionRightEye
  - Tobii.Research.Addons.CalibrationValidationPoint, [84](#)
- PrecisionRmsLeft
  - GazeUtilityLibrary.DataStructs.GazeValidationData, [146](#)
- PrecisionRMSLeftEye
  - Tobii.Research.Addons.CalibrationValidationPoint, [84](#)
- PrecisionRmsRight
  - GazeUtilityLibrary.DataStructs.GazeValidationData, [146](#)
- PrecisionRMSRightEye
  - Tobii.Research.Addons.CalibrationValidationPoint, [84](#)
- Prepare
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [122](#)
  - GazeUtilityLibrary.DataStructs.GazeData, [133](#)
  - GazeUtilityLibrary.DataStructs.GazeValidationPoint, [148](#)
- PrepareCalibrationOutputFile
  - GazeUtilityLibrary.GazeConfiguration, [128](#)
- PrepareGazeOutputFile
  - GazeUtilityLibrary.GazeConfiguration, [129](#)
- PrepareValidationOutputFile
  - GazeUtilityLibrary.GazeConfiguration, [129](#)
- PropertyChanged
  - CustomCalibrationLibrary.Models.CalibrationModel, [69](#)
  - CustomCalibrationLibrary.Views.CalibrationFailed, [61](#)
  - GazeUtilityLibrary.DataStructs.CalibrationPoint, [73](#)
  - GazeUtilityLibrary.DataStructs.LiveGazePoint, [154](#)
  - GazeUtilityLibrary.DataStructs.UserPositionData, [193](#)
  - GazeUtilityLibrary.Tracker.BaseTracker, [52](#)
- PupilDiameter
  - GazeUtilityLibrary.DataStructs.EyeData, [112](#)
- Q
  - GazeUtilityLibrary.DriftCompensation, [107](#)
- ReadyTimer
  - GazeUtilityLibrary.ConfigItem, [98](#)
- RedoCalibrationPoint
  - CustomCalibrationLibrary.Models.CalibrationModel, [66](#)
- Reset
  - GazeUtilityLibrary.DriftCompensation, [106](#)
- ResetDriftCompensation
  - GazeToMouse.App, [35](#)
  - GazeUtilityLibrary.Tracker.BaseTracker, [50](#)
- ResetStartTime
  - GazeUtilityLibrary.DataStructs.PipeCommand, [165](#)
- Result
  - GazeUtilityLibrary.DataStructs.GazeValidationPoint, [148](#)
- Tobii.Research.Addons.ScreenBasedCalibrationValidation, [180](#)
- Right
  - GazeUtilityLibrary.DataStructs.GazeData, [134](#)
- ScreenArea
  - GazeUtilityLibrary.ConfigItem, [98](#)
  - GazeUtilityLibrary.ScreenArea, [172](#)
  - GazeUtilityLibrary.Tracker.BaseTracker, [52](#)
- screenArea
  - GazeUtilityLibrary.Tracker.BaseTracker, [51](#)
- ScreenBasedCalibrationValidation
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, [178](#)
- ScreenSelection
  - CustomCalibrationLibrary.Views.ScreenSelection, [182](#)
- ScreenSelectionViewModel
  - CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel, [183](#)
- ScreenSwitchCommand
  - CustomCalibrationLibrary.ViewModels.ScreenSelectionViewModel, [184](#)
- ScreenTriangle
  - GazeUtilityLibrary.ScreenTriangle, [185](#)
- SerializeJsonConfig
  - GazeUtilityLibrary.JsonConfigParser, [152](#)
- SetCalibrationResult
  - CustomCalibrationLibrary.Models.CalibrationModel, [66](#)
- ShowCursor
  - GazeUtilityLibrary.MouseHider, [157](#)
- ShowMouse, [30](#)
- ShowMouse.App, [36](#)
- Start
  - GazeUtilityLibrary.DriftCompensation, [106](#)
  - GazeUtilityLibrary.Tracker.MouseTracker, [163](#)
- StartCollectingData
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, [180](#)
- StartDriftCompensation
  - GazeUtilityLibrary.Tracker.BaseTracker, [50](#)
- StartTime
  - GazeToMouse.App, [35](#)
- State
  - GazeUtilityLibrary.Tracker.BaseTracker, [52](#)
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, [180](#)
- Status
  - CustomCalibrationLibrary.Models.CalibrationModel, [68](#)
- Stop
  - GazeUtilityLibrary.Tracker.MouseTracker, [164](#)
- Tag
  - GazeToMouse.App, [35](#)
- TimedOut
  - Tobii.Research.Addons.CalibrationValidationPoint, [84](#)



- Timestamp
  - GazeUtilityLibrary.DataStructs.GazeData, 134
- TimestampReceived
  - GazeUtilityLibrary.DataStructs.GazeData, 134
- Tobii, 30
- Tobii.Research, 30
- Tobii.Research.Addons, 30
- Tobii.Research.Addons.CalibrationValidationPoint, 82
  - AccuracyLeftEye, 83
  - AccuracyRightEye, 83
  - Coordinates, 83
  - GazeData, 83
  - PrecisionLeftEye, 84
  - PrecisionRightEye, 84
  - PrecisionRMSLeftEye, 84
  - PrecisionRMSRightEye, 84
  - TimedOut, 84
  - ToString, 83
- Tobii.Research.Addons.CalibrationValidationResult, 85
  - AverageAccuracyLeftEye, 86
  - AverageAccuracyRightEye, 86
  - AveragePrecisionLeftEye, 86
  - AveragePrecisionRightEye, 86
  - AveragePrecisionRMSLeftEye, 86
  - AveragePrecisionRMSRightEye, 86
  - Points, 87
  - ToString, 85
- Tobii.Research.Addons.ScreenBasedCalibrationValidation, 176
  - Compute, 179
  - DiscardData, 179
  - Dispose, 179
  - EnterValidationMode, 179
  - LeaveValidationMode, 179
  - Result, 180
  - ScreenBasedCalibrationValidation, 178
  - StartCollectingData, 180
  - State, 180
  - ToString, 180
  - ValidationState, 178
- Tobii.Research.Addons.Utility, 30
- TobiiApplicationPath
  - GazeUtilityLibrary.ConfigItem, 98
- TobiiCalibrate, 30
  - GazeUtilityLibrary.ConfigItem, 99
- TobiiCalibrate.App, 37
- TobiiCalibrateArguments
  - GazeUtilityLibrary.ConfigItem, 99
- TopLeft
  - GazeUtilityLibrary.ConfigScreenArea, 102
  - GazeUtilityLibrary.ScreenArea, 175
- TopRight
  - GazeUtilityLibrary.ConfigScreenArea, 102
  - GazeUtilityLibrary.ScreenArea, 176
- ToString
  - Tobii.Research.Addons.CalibrationValidationPoint, 83
  - Tobii.Research.Addons.CalibrationValidationResult, 85
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, 180
- TrackerDevice
  - GazeUtilityLibrary.ConfigItem, 99
- TrackerDisabled
  - GazeUtilityLibrary.Tracker.BaseTracker, 52
- TrackerEnabled
  - GazeUtilityLibrary.Tracker.BaseTracker, 53
- TrackerLogger
  - GazeUtilityLibrary.TrackerLogger, 187
- trackerMessageBox
  - GazeUtilityLibrary.Tracker.BaseTracker, 51
- TrialId
  - GazeToMouse.App, 36
  - GazeUtilityLibrary.DataStructs.PipeCommand, 165
- Update
  - GazeUtilityLibrary.DriftCompensation, 106
- UpdateGazePoint
  - CustomCalibrationLibrary.Models.CalibrationModel, 67
- UserPosition
  - CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel, 196
- UserPositionData
  - GazeUtilityLibrary.DataStructs.UserPositionData, 191
- UserPositionDataHandler
  - GazeUtilityLibrary.Tracker.BaseTracker, 50
- UserPositionDataReceived
  - GazeUtilityLibrary.Tracker.BaseTracker, 53
- UserPositionGuide
  - CustomCalibrationLibrary.Models.CalibrationModel, 68
  - CustomCalibrationLibrary.Views.UserPositionGuide, 194
- UserPositionGuideChanged
  - CustomCalibrationLibrary.Models.CalibrationModel, 69
- UserPositionGuideViewModel
  - CustomCalibrationLibrary.ViewModels.UserPositionGuideViewModel, 195
- V1
  - GazeUtilityLibrary.ScreenTriangle, 186
- V2
  - GazeUtilityLibrary.ScreenTriangle, 186
- V3
  - GazeUtilityLibrary.ScreenTriangle, 186
- ValidationCloseCommand
  - CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, 198
- ValidationData
  - CustomCalibrationLibrary.Models.CalibrationModel, 68
  - CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, 199

- ValidationDurationThreshold
  - GazeUtilityLibrary.ConfigItem, [99](#)
- ValidationLogColumnOrder
  - GazeUtilityLibrary.ConfigItem, [99](#)
- ValidationLogColumnName
  - GazeUtilityLibrary.ConfigItem, [99](#)
- ValidationLogWriteOutput
  - GazeUtilityLibrary.ConfigItem, [100](#)
- ValidationOutputValue
  - GazeUtilityLibrary.DataStructs, [29](#)
- ValidationPoints
  - GazeUtilityLibrary.ConfigItem, [100](#)
- ValidationRestartCommand
  - CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, [199](#)
- ValidationResult
  - CustomCalibrationLibrary.Views.ValidationResult, [197](#)
- ValidationResultViewModel
  - CustomCalibrationLibrary.ViewModels.ValidationResultViewModel, [198](#)
- ValidationState
  - Tobii.Research.Addons.ScreenBasedCalibrationValidation, [178](#)
- ValidationTimer
  - GazeUtilityLibrary.ConfigItem, [100](#)
- ValidityLeft
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [122](#)
- ValidityRight
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [122](#)
- Visibility
  - GazeUtilityLibrary.DataStructs.LiveGazePoint, [154](#)
- Warning
  - GazeUtilityLibrary.TrackerLogger, [188](#)
- Width
  - GazeUtilityLibrary.ConfigScreenArea, [103](#)
  - GazeUtilityLibrary.ScreenArea, [176](#)
- WriteToCalibrationOutput
  - GazeUtilityLibrary.GazeConfiguration, [129](#)
- WriteToGazeOutput
  - GazeUtilityLibrary.GazeConfiguration, [130](#)
- WriteToValidationOutput
  - GazeUtilityLibrary.GazeConfiguration, [130](#)
- X
  - GazeUtilityLibrary.DataStructs.LiveGazePoint, [154](#)
- XCoord
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [122](#)
- XCoordLeft
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [122](#)
  - GazeUtilityLibrary.DataStructs.UserPositionData, [192](#)
- XCoordRight
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [123](#)
  - GazeUtilityLibrary.DataStructs.UserPositionData, [192](#)
- Y
  - GazeUtilityLibrary.DataStructs.LiveGazePoint, [154](#)
- YCoord
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [123](#)
- YCoordLeft
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [123](#)
  - GazeUtilityLibrary.DataStructs.UserPositionData, [192](#)
- YCoordRight
  - GazeUtilityLibrary.DataStructs.GazeCalibrationData, [123](#)
  - GazeUtilityLibrary.DataStructs.UserPositionData, [192](#)
- ZCoordLeft
  - GazeUtilityLibrary.DataStructs.UserPositionData, [192](#)
- ZCoordRight
  - GazeUtilityLibrary.DataStructs.UserPositionData, [193](#)