

## Reference Manual Signapse - Traffic Sign Detection

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	Package de . . . . .	9
5.2	Package de.swp . . . . .	9
5.3	Package de.swp.tsd . . . . .	9
5.4	Package de.swp.tsd.trafficsigndetection . . . . .	9
<b>6</b>	<b>Class Documentation</b>	<b>11</b>
6.1	de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback Class Reference . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.1.2	Constructor & Destructor Documentation . . . . .	11
6.1.2.1	DiffCallback() . . . . .	12
6.1.3	Member Function Documentation . . . . .	12
6.1.3.1	areContentsTheSame() . . . . .	12
6.1.3.2	areItemsTheSame() . . . . .	12

6.1.3.3	<a href="#">getChangePayload()</a>	13
6.1.3.4	<a href="#">getNewListSize()</a>	13
6.1.3.5	<a href="#">getOldListSize()</a>	14
6.2	<a href="#">de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.FrameEventListener Interface Reference</a>	14
6.2.1	<a href="#">Detailed Description</a>	14
6.2.2	<a href="#">Member Function Documentation</a>	14
6.2.2.1	<a href="#">onFrameFinished()</a>	14
6.3	<a href="#">de.swp.tsd.trafficsigndetection.MainActivity Class Reference</a>	15
6.3.1	<a href="#">Detailed Description</a>	16
6.3.2	<a href="#">Member Function Documentation</a>	16
6.3.2.1	<a href="#">[static initializer]()</a>	16
6.3.2.2	<a href="#">getContentLayout()</a>	16
6.3.2.3	<a href="#">onBackPressed()</a>	17
6.3.2.4	<a href="#">onClickMenu()</a>	17
6.3.2.5	<a href="#">onCreate()</a>	17
6.3.2.6	<a href="#">onDisclaimerAccepted()</a>	17
6.3.2.7	<a href="#">onDisclosureAccepted()</a>	18
6.3.2.8	<a href="#">onPrivacyPolicyAccepted()</a>	18
6.3.2.9	<a href="#">onRequestPermissionsResult()</a>	18
6.3.2.10	<a href="#">openDetectorFragment()</a>	19
6.4	<a href="#">de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener Interface Reference</a>	19
6.4.1	<a href="#">Detailed Description</a>	19
6.4.2	<a href="#">Member Function Documentation</a>	20
6.4.2.1	<a href="#">onClose()</a>	20
6.4.2.2	<a href="#">onOpened()</a>	20
6.4.2.3	<a href="#">onPaused()</a>	20
6.4.2.4	<a href="#">onResume()</a>	20
6.5	<a href="#">de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.OnCutOffSpeedCalculatedListener Interface Reference</a>	20
6.5.1	<a href="#">Detailed Description</a>	21

6.5.2	Member Function Documentation	21
6.5.2.1	onCutOffSpeedCalculated()	21
6.6	de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionResultListener Interface Reference	21
6.6.1	Detailed Description	21
6.6.2	Member Function Documentation	22
6.6.2.1	getDeclineText()	22
6.6.2.2	getRequestedPermissions()	22
6.6.2.3	onRequestedPermissionsDeclined()	22
6.6.2.4	onRequestedPermissionsGranted()	22
6.7	de.swp.tsd.trafficsigndetection.TSDRotationSensor.OnRotationListener Interface Reference	23
6.7.1	Detailed Description	23
6.7.2	Member Function Documentation	23
6.7.2.1	onRotation()	23
6.8	de.swp.tsd.trafficsigndetection.TSDSpeedTracker.OnSpeedCalculatedListener Interface Reference	23
6.8.1	Detailed Description	23
6.8.2	Member Function Documentation	24
6.8.2.1	onSpeedCalculated()	24
6.9	de.swp.tsd.trafficsigndetection.TSDAboutFragment Class Reference	24
6.9.1	Detailed Description	24
6.9.2	Constructor & Destructor Documentation	25
6.9.2.1	TSDAboutFragment()	25
6.9.3	Member Function Documentation	25
6.9.3.1	onCreateView()	25
6.10	de.swp.tsd.trafficsigndetection.TSDAutoFocusMode Enum Reference	25
6.10.1	Detailed Description	26
6.10.2	Constructor & Destructor Documentation	26
6.10.2.1	TSDAutoFocusMode()	26
6.10.3	Member Function Documentation	26
6.10.3.1	fromInt()	26
6.10.4	Member Data Documentation	27

6.10.4.1	AUTO	27
6.10.4.2	CONTINUOUS_PICTURE	27
6.10.4.3	CONTINUOUS_VIDEO	27
6.10.4.4	EDOF	27
6.10.4.5	MACRO	27
6.10.4.6	OFF	28
6.10.4.7	value	28
6.11	de.swp.tsd.trafficsigndetection.TSDBaseCamera Class Reference	28
6.11.1	Detailed Description	29
6.11.2	Constructor & Destructor Documentation	29
6.11.2.1	TSDBaseCamera()	29
6.11.3	Member Function Documentation	29
6.11.3.1	getBackgroundHandler()	30
6.11.3.2	getCameraStateCallback()	30
6.11.3.3	getCharacter()	30
6.11.3.4	onPause()	30
6.11.3.5	onResume()	31
6.11.3.6	onTargetsReady()	31
6.12	de.swp.tsd.trafficsigndetection.TSDCameraCharacter Class Reference	31
6.12.1	Detailed Description	32
6.12.2	Constructor & Destructor Documentation	32
6.12.2.1	TSDCameraCharacter()	32
6.12.3	Member Function Documentation	32
6.12.3.1	getAvailableApertures()	32
6.12.3.2	getAvailableAutoFocusModes()	33
6.12.3.3	getAvailableOisModes()	33
6.12.3.4	getAvailableOutputFormats()	33
6.12.3.5	getAvailableOutputSizes()	33
6.12.3.6	getCameraCharacteristics()	34
6.12.3.7	getId()	34

6.12.3.8	<a href="#">getLensFacing()</a>	34
6.12.3.9	<a href="#">getManager()</a>	35
6.12.3.10	<a href="#">getMatchingOutputSize()</a>	35
6.12.3.11	<a href="#">getMaximumViewAngle()</a>	35
6.12.3.12	<a href="#">getScore()</a>	36
6.12.3.13	<a href="#">getSensorRotation()</a>	36
6.13	<a href="#">de.swp.tsd.trafficsigndetection.TSDCameraManager Class Reference</a>	36
6.13.1	<a href="#">Detailed Description</a>	37
6.13.2	<a href="#">Constructor &amp; Destructor Documentation</a>	37
6.13.2.1	<a href="#">TSDCameraManager()</a>	37
6.13.3	<a href="#">Member Function Documentation</a>	37
6.13.3.1	<a href="#">getCameraCharacters()</a>	38
6.13.3.2	<a href="#">openCamera()</a>	38
6.14	<a href="#">de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat Enum Reference</a>	38
6.14.1	<a href="#">Detailed Description</a>	39
6.14.2	<a href="#">Constructor &amp; Destructor Documentation</a>	39
6.14.2.1	<a href="#">TSDCameraOutputFormat()</a>	39
6.14.3	<a href="#">Member Function Documentation</a>	40
6.14.3.1	<a href="#">fromInt()</a>	40
6.14.4	<a href="#">Member Data Documentation</a>	40
6.14.4.1	<a href="#">A_8</a>	40
6.14.4.2	<a href="#">DEPTH16</a>	40
6.14.4.3	<a href="#">DEPTH_POINT_CLOUD</a>	40
6.14.4.4	<a href="#">FLEX_RGB_888</a>	41
6.14.4.5	<a href="#">FLEX_RGBA_8888</a>	41
6.14.4.6	<a href="#">GBA_5551</a>	41
6.14.4.7	<a href="#">JPEG</a>	41
6.14.4.8	<a href="#">L_8</a>	41
6.14.4.9	<a href="#">LA_88</a>	41
6.14.4.10	<a href="#">NV16</a>	41

6.14.4.11 NV21 . . . . .	41
6.14.4.12 OPAQUE . . . . .	42
6.14.4.13 PRIVATE . . . . .	42
6.14.4.14 RAW10 . . . . .	42
6.14.4.15 RAW12 . . . . .	42
6.14.4.16 RAW_PRIVATE . . . . .	42
6.14.4.17 RAW_SENSOR . . . . .	42
6.14.4.18 RGB_332 . . . . .	42
6.14.4.19 RGB_565 . . . . .	42
6.14.4.20 RGB_888 . . . . .	43
6.14.4.21 RGBA_1010102 . . . . .	43
6.14.4.22 RGBA_4444 . . . . .	43
6.14.4.23 RGBA_8888 . . . . .	43
6.14.4.24 RGBA_F16 . . . . .	43
6.14.4.25 RGBX_8888 . . . . .	43
6.14.4.26 TRANSLUCENT . . . . .	43
6.14.4.27 TRANSPARENT . . . . .	43
6.14.4.28 UNKNOWN . . . . .	44
6.14.4.29 value . . . . .	44
6.14.4.30 YUV_420_888 . . . . .	44
6.14.4.31 YUV_422_888 . . . . .	44
6.14.4.32 YUV_444_888 . . . . .	44
6.14.4.33 YUY2 . . . . .	44
6.14.4.34 YV12 . . . . .	44
6.15 de.swp.tsd.trafficsign detection.TSDCutOffSpeedTracker Class Reference . . . . .	45
6.15.1 Detailed Description . . . . .	45
6.15.2 Constructor & Destructor Documentation . . . . .	45
6.15.2.1 TSDCutOffSpeedTracker() . . . . .	45
6.15.3 Member Function Documentation . . . . .	46
6.15.3.1 onFrameFinished() . . . . .	46



6.15.3.2	<a href="#">setCameraViewAngle()</a>	46
6.16	<a href="#">de.swp.tsd.trafficsigndetection.TSDDebugFragment Class Reference</a>	47
6.16.1	Detailed Description	47
6.16.2	Constructor & Destructor Documentation	47
6.16.2.1	<a href="#">TSDDebugFragment()</a>	48
6.16.3	Member Function Documentation	48
6.16.3.1	<a href="#">onCreate()</a>	48
6.16.3.2	<a href="#">onCreateView()</a>	48
6.16.3.3	<a href="#">onStart()</a>	49
6.17	<a href="#">de.swp.tsd.trafficsigndetection.TSDDetectorFragment Class Reference</a>	49
6.17.1	Detailed Description	50
6.17.2	Constructor & Destructor Documentation	50
6.17.2.1	<a href="#">TSDDetectorFragment()</a>	50
6.17.3	Member Function Documentation	50
6.17.3.1	<a href="#">onCreate()</a>	50
6.17.3.2	<a href="#">onCreateView()</a>	51
6.17.3.3	<a href="#">onStart()</a>	52
6.18	<a href="#">de.swp.tsd.trafficsigndetection.TSDDisclaimerFragment Class Reference</a>	52
6.18.1	Detailed Description	52
6.18.2	Constructor & Destructor Documentation	52
6.18.2.1	<a href="#">TSDDisclaimerFragment()</a>	53
6.18.3	Member Function Documentation	53
6.18.3.1	<a href="#">onCreateView()</a>	53
6.19	<a href="#">de.swp.tsd.trafficsigndetection.TSDDisclosureFragment Class Reference</a>	53
6.19.1	Detailed Description	54
6.19.2	Constructor & Destructor Documentation	54
6.19.2.1	<a href="#">TSDDisclosureFragment()</a>	54
6.19.3	Member Function Documentation	54
6.19.3.1	<a href="#">onCreateView()</a>	54
6.20	<a href="#">de.swp.tsd.trafficsigndetection.TSDDrawable Class Reference</a>	55

6.20.1 Detailed Description . . . . .	55
6.20.2 Constructor & Destructor Documentation . . . . .	55
6.20.2.1 TSDDrawable() [1/2] . . . . .	55
6.20.2.2 TSDDrawable() [2/2] . . . . .	56
6.20.3 Member Function Documentation . . . . .	56
6.20.3.1 equals() . . . . .	56
6.20.3.2 getDrawable() . . . . .	57
6.20.3.3 getDrawableId() . . . . .	57
6.20.3.4 getHeight() . . . . .	57
6.20.3.5 getWidth() . . . . .	57
6.21 de.swp.tsd.trafficsigndetection.TSDDrawableDatabase Class Reference . . . . .	58
6.21.1 Detailed Description . . . . .	58
6.21.2 Member Function Documentation . . . . .	58
6.21.2.1 getDrawable() . . . . .	58
6.22 de.swp.tsd.trafficsigndetection.TSDFragment Class Reference . . . . .	59
6.22.1 Detailed Description . . . . .	59
6.22.2 Constructor & Destructor Documentation . . . . .	59
6.22.2.1 TSDFragment() . . . . .	60
6.22.3 Member Data Documentation . . . . .	61
6.22.3.1 displayIconId . . . . .	61
6.22.3.2 displayNameId . . . . .	61
6.23 de.swp.tsd.trafficsigndetection.TSDHelper Class Reference . . . . .	61
6.23.1 Detailed Description . . . . .	61
6.23.2 Member Function Documentation . . . . .	62
6.23.2.1 dpToPx() . . . . .	62
6.23.2.2 equalNullable() . . . . .	63
6.23.2.3 readBooleanFromSharedPreferences() . . . . .	63
6.23.2.4 readIntegerFromSharedPreferences() . . . . .	64
6.23.2.5 writeBooleanToSharedPreferences() . . . . .	64
6.23.2.6 writeIntegerToSharedPreferences() . . . . .	64

6.24	<a href="#">de.swp.tsd.trafficsigndetection.TSDInitializationErrorDialog Class Reference</a>	65
6.24.1	<a href="#">Detailed Description</a>	65
6.24.2	<a href="#">Member Function Documentation</a>	65
6.24.2.1	<a href="#">onCreateDialog()</a>	66
6.25	<a href="#">de.swp.tsd.trafficsigndetection.TSDLensFacing Enum Reference</a>	66
6.25.1	<a href="#">Detailed Description</a>	66
6.25.2	<a href="#">Constructor &amp; Destructor Documentation</a>	66
6.25.2.1	<a href="#">TSDLensFacing()</a>	66
6.25.3	<a href="#">Member Function Documentation</a>	67
6.25.3.1	<a href="#">fromInt()</a>	67
6.25.4	<a href="#">Member Data Documentation</a>	67
6.25.4.1	<a href="#">BACK</a>	67
6.25.4.2	<a href="#">EXTERNAL</a>	67
6.25.4.3	<a href="#">FRONT</a>	67
6.25.4.4	<a href="#">value</a>	68
6.26	<a href="#">de.swp.tsd.trafficsigndetection.TSDLicenseFragment Class Reference</a>	68
6.26.1	<a href="#">Detailed Description</a>	68
6.26.2	<a href="#">Constructor &amp; Destructor Documentation</a>	68
6.26.2.1	<a href="#">TSDLicenseFragment()</a>	68
6.26.3	<a href="#">Member Function Documentation</a>	69
6.26.3.1	<a href="#">onCreateView()</a>	69
6.27	<a href="#">de.swp.tsd.trafficsigndetection.TSDLifecycleObserver Class Reference</a>	69
6.27.1	<a href="#">Detailed Description</a>	70
6.27.2	<a href="#">Constructor &amp; Destructor Documentation</a>	70
6.27.2.1	<a href="#">TSDLifecycleObserver()</a>	70
6.27.3	<a href="#">Member Function Documentation</a>	70
6.27.3.1	<a href="#">betweenResumeAndPause()</a>	70
6.27.3.2	<a href="#">disable()</a>	71
6.27.3.3	<a href="#">enable()</a>	71
6.27.3.4	<a href="#">isEnabled()</a>	71

6.27.3.5	<a href="#">onPause()</a>	71
6.27.3.6	<a href="#">onResume()</a>	71
6.27.3.7	<a href="#">onStart()</a>	72
6.28	<a href="#">de.swp.tsd.trafficsigndetection.TSDLocationListener Class Reference</a>	72
6.28.1	<a href="#">Detailed Description</a>	72
6.28.2	<a href="#">Constructor &amp; Destructor Documentation</a>	72
6.28.2.1	<a href="#">TSDLocationListener()</a>	72
6.28.3	<a href="#">Member Function Documentation</a>	73
6.28.3.1	<a href="#">onLocationChanged()</a>	73
6.28.3.2	<a href="#">onProviderDisabled()</a>	73
6.28.3.3	<a href="#">onProviderEnabled()</a>	73
6.28.3.4	<a href="#">onStatusChanged()</a>	74
6.29	<a href="#">de.swp.tsd.trafficsigndetection.TSDNavigationAdapter Class Reference</a>	74
6.29.1	<a href="#">Detailed Description</a>	75
6.29.2	<a href="#">Constructor &amp; Destructor Documentation</a>	75
6.29.2.1	<a href="#">TSDNavigationAdapter()</a>	75
6.29.3	<a href="#">Member Function Documentation</a>	75
6.29.3.1	<a href="#">getCount()</a>	75
6.29.3.2	<a href="#">getItem()</a>	75
6.29.3.3	<a href="#">getItemId()</a>	76
6.29.3.4	<a href="#">getView()</a>	76
6.30	<a href="#">de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode Enum Reference</a>	77
6.30.1	<a href="#">Detailed Description</a>	77
6.30.2	<a href="#">Constructor &amp; Destructor Documentation</a>	77
6.30.2.1	<a href="#">TSDOpticalStabilizationMode()</a>	77
6.30.3	<a href="#">Member Function Documentation</a>	78
6.30.3.1	<a href="#">fromInt()</a>	78
6.30.4	<a href="#">Member Data Documentation</a>	78
6.30.4.1	<a href="#">OFF</a>	78
6.30.4.2	<a href="#">ON</a>	78

6.30.4.3	value	78
6.31	de.swp.tsd.trafficsigndetection.TSDPeriodicTask Class Reference	79
6.31.1	Detailed Description	79
6.31.2	Constructor & Destructor Documentation	79
6.31.2.1	TSDPeriodicTask()	79
6.31.3	Member Function Documentation	80
6.31.3.1	onPause()	80
6.31.3.2	onResume()	80
6.32	de.swp.tsd.trafficsigndetection.TSDPermissionManager Class Reference	80
6.32.1	Detailed Description	81
6.32.2	Member Function Documentation	81
6.32.2.1	getPermissions()	81
6.32.2.2	onRequestPermissionResult()	81
6.33	de.swp.tsd.trafficsigndetection.TSDPrivacyFragment Class Reference	82
6.33.1	Detailed Description	82
6.33.2	Constructor & Destructor Documentation	83
6.33.2.1	TSDPrivacyFragment()	83
6.33.3	Member Function Documentation	83
6.33.3.1	onCreate()	83
6.33.3.2	onCreateView()	83
6.34	de.swp.tsd.trafficsigndetection.TSDRingMemory< T > Class Template Reference	84
6.34.1	Detailed Description	84
6.34.2	Constructor & Destructor Documentation	84
6.34.2.1	TSDRingMemory()	85
6.34.3	Member Function Documentation	85
6.34.3.1	add() [1/2]	85
6.34.3.2	add() [2/2]	85
6.34.3.3	addAll() [1/2]	86
6.34.3.4	addAll() [2/2]	86
6.34.3.5	getAllSkipLast()	86

6.34.3.6	<a href="#">getMaxSize()</a>	87
6.34.3.7	<a href="#">setMaxSize()</a>	87
6.35	<a href="#">de.swp.tsd.trafficsigndetection.TSDRotationSensor Class Reference</a>	87
6.35.1	<a href="#">Detailed Description</a>	88
6.35.2	<a href="#">Constructor &amp; Destructor Documentation</a>	88
6.35.2.1	<a href="#">TSDRotationSensor()</a>	88
6.35.3	<a href="#">Member Function Documentation</a>	88
6.35.3.1	<a href="#">onPause()</a>	89
6.35.3.2	<a href="#">onResume()</a>	89
6.36	<a href="#">de.swp.tsd.trafficsigndetection.TSDSettingsFragment Class Reference</a>	89
6.36.1	<a href="#">Detailed Description</a>	89
6.36.2	<a href="#">Constructor &amp; Destructor Documentation</a>	89
6.36.2.1	<a href="#">TSDSettingsFragment()</a>	90
6.36.3	<a href="#">Member Function Documentation</a>	90
6.36.3.1	<a href="#">onCreate()</a>	90
6.36.3.2	<a href="#">onCreateView()</a>	90
6.37	<a href="#">de.swp.tsd.trafficsigndetection.TSDSetupFragment Class Reference</a>	91
6.37.1	<a href="#">Detailed Description</a>	91
6.37.2	<a href="#">Constructor &amp; Destructor Documentation</a>	91
6.37.2.1	<a href="#">TSDSetupFragment()</a>	92
6.37.3	<a href="#">Member Function Documentation</a>	92
6.37.3.1	<a href="#">onCreate()</a>	92
6.37.3.2	<a href="#">onCreateView()</a>	92
6.37.3.3	<a href="#">onStart()</a>	93
6.38	<a href="#">de.swp.tsd.trafficsigndetection.TSDShutterCamera Class Reference</a>	93
6.38.1	<a href="#">Detailed Description</a>	93
6.38.2	<a href="#">Constructor &amp; Destructor Documentation</a>	94
6.38.2.1	<a href="#">TSDShutterCamera()</a>	94
6.38.3	<a href="#">Member Function Documentation</a>	94
6.38.3.1	<a href="#">disable()</a>	94

6.38.3.2	<a href="#">enable()</a>	94
6.38.3.3	<a href="#">getCameraCharacter()</a>	95
6.38.3.4	<a href="#">getSensorRotation()</a>	95
6.38.3.5	<a href="#">isEnabled()</a>	95
6.38.3.6	<a href="#">isInitiated()</a>	95
6.39	<a href="#">de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener Class Reference</a>	96
6.39.1	<a href="#">Detailed Description</a>	96
6.39.2	<a href="#">Constructor &amp; Destructor Documentation</a>	96
6.39.2.1	<a href="#">TSDShutterCameraImageListener()</a>	96
6.39.3	<a href="#">Member Function Documentation</a>	97
6.39.3.1	<a href="#">onImageAvailable()</a>	97
6.39.3.2	<a href="#">roadSignAPIfeedImage()</a>	97
6.39.3.3	<a href="#">setSensorRotation()</a>	98
6.40	<a href="#">de.swp.tsd.trafficsigndetection.TSDSign Enum Reference</a>	98
6.40.1	<a href="#">Detailed Description</a>	100
6.40.2	<a href="#">Constructor &amp; Destructor Documentation</a>	100
6.40.2.1	<a href="#">TSDSign()</a>	100
6.40.3	<a href="#">Member Function Documentation</a>	100
6.40.3.1	<a href="#">endsZone()</a>	100
6.40.3.2	<a href="#">fromId()</a>	101
6.40.3.3	<a href="#">getDrawable()</a>	101
6.40.3.4	<a href="#">isRelevant()</a>	102
6.40.3.5	<a href="#">isStandalone()</a>	102
6.40.3.6	<a href="#">isZoneStart()</a>	102
6.40.4	<a href="#">Member Data Documentation</a>	102
6.40.4.1	<a href="#">MISC</a>	103
6.40.4.2	<a href="#">MISC_BLUE</a>	103
6.40.4.3	<a href="#">MISC_BROWN</a>	103
6.40.4.4	<a href="#">MISC_GREEN</a>	103
6.40.4.5	<a href="#">MISC_RED_BLUE</a>	103

6.40.4.6	MISC_RED_WHITE	103
6.40.4.7	MISC_WHITE	103
6.40.4.8	MISC_YELLOW	103
6.40.4.9	speedLimit	104
6.40.4.10	VZ1000	104
6.40.4.11	VZ1001_30	104
6.40.4.12	VZ1001_31	104
6.40.4.13	VZ1001_32	104
6.40.4.14	VZ1001_33	104
6.40.4.15	VZ1004_30	104
6.40.4.16	VZ1004_31	105
6.40.4.17	VZ101	105
6.40.4.18	VZ101_11	105
6.40.4.19	VZ103_10	105
6.40.4.20	VZ103_20	105
6.40.4.21	VZ1040_30	105
6.40.4.22	VZ1042_31	105
6.40.4.23	VZ1042_33	105
6.40.4.24	VZ1053_35	106
6.40.4.25	VZ105_10	106
6.40.4.26	VZ105_20	106
6.40.4.27	VZ120	106
6.40.4.28	VZ121_10	106
6.40.4.29	VZ121_20	106
6.40.4.30	VZ123	106
6.40.4.31	VZ131	106
6.40.4.32	VZ133_10	107
6.40.4.33	VZ136_10	107
6.40.4.34	VZ274_10	107
6.40.4.35	VZ274_100	107



6.40.4.36 VZ274_110 . . . . .	107
6.40.4.37 VZ274_120 . . . . .	107
6.40.4.38 VZ274_130 . . . . .	107
6.40.4.39 VZ274_20 . . . . .	107
6.40.4.40 VZ274_30 . . . . .	108
6.40.4.41 VZ274_40 . . . . .	108
6.40.4.42 VZ274_50 . . . . .	108
6.40.4.43 VZ274_60 . . . . .	108
6.40.4.44 VZ274_70 . . . . .	108
6.40.4.45 VZ274_80 . . . . .	108
6.40.4.46 VZ274_90 . . . . .	108
6.40.4.47 VZ274x1 . . . . .	108
6.40.4.48 VZ274x2 . . . . .	109
6.40.4.49 VZ278 . . . . .	109
6.40.4.50 VZ282 . . . . .	109
6.40.4.51 VZ306 . . . . .	109
6.40.4.52 VZ310 . . . . .	109
6.40.4.53 VZ311 . . . . .	109
6.40.4.54 VZ325x1 . . . . .	109
6.40.4.55 VZ325x2 . . . . .	109
6.40.4.56 VZ330x1 . . . . .	110
6.40.4.57 VZ330x2 . . . . .	110
6.40.4.58 VZ331x1 . . . . .	110
6.40.4.59 VZ331x2 . . . . .	110
6.40.4.60 VZ523_30 . . . . .	110
6.41 de.swp.tsd.trafficsigndetection.TSDSignCombination Class Reference . . . . .	110
6.41.1 Detailed Description . . . . .	111
6.41.2 Constructor & Destructor Documentation . . . . .	111
6.41.2.1 TSDSignCombination() . . . . .	111
6.41.3 Member Function Documentation . . . . .	112

6.41.3.1	<a href="#">createUpdatedBySublist()</a>	112
6.41.3.2	<a href="#">equals()</a>	112
6.41.3.3	<a href="#">getDistanceToImageCenter()</a>	113
6.41.3.4	<a href="#">getFrameId()</a>	113
6.41.3.5	<a href="#">getId()</a>	113
6.41.3.6	<a href="#">getSignDrawables()</a>	114
6.41.3.7	<a href="#">getSigns()</a>	114
6.41.3.8	<a href="#">getSpeed()</a>	114
6.41.3.9	<a href="#">getTimestamp()</a>	114
6.41.3.10	<a href="#">isStandalone()</a>	115
6.41.3.11	<a href="#">isSublistOf()</a>	115
6.41.3.12	<a href="#">setSpeed()</a>	115
6.42	<a href="#">de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder Class Reference</a>	115
6.42.1	<a href="#">Detailed Description</a>	116
6.42.2	<a href="#">Constructor &amp; Destructor Documentation</a>	116
6.42.2.1	<a href="#">TSDSignCombinationViewHolder()</a>	116
6.42.3	<a href="#">Member Function Documentation</a>	116
6.42.3.1	<a href="#">displayAll()</a>	117
6.42.3.2	<a href="#">displayAllChanges()</a>	117
6.42.3.3	<a href="#">displaySigns()</a>	118
6.42.3.4	<a href="#">displaySpeed()</a>	118
6.42.3.5	<a href="#">displayTime()</a>	118
6.42.3.6	<a href="#">setLeftRightMargin()</a>	119
6.42.3.7	<a href="#">setWarningTransparency()</a>	119
6.43	<a href="#">de.swp.tsd.trafficsigndetection.TSDSignHistory Class Reference</a>	119
6.43.1	<a href="#">Detailed Description</a>	120
6.43.2	<a href="#">Constructor &amp; Destructor Documentation</a>	120
6.43.2.1	<a href="#">TSDSignHistory()</a>	120
6.43.3	<a href="#">Member Function Documentation</a>	120
6.43.3.1	<a href="#">addSignCombination()</a>	120

6.43.3.2	<a href="#">getLatestFrameId()</a>	121
6.43.3.3	<a href="#">getLatestId()</a>	121
6.43.3.4	<a href="#">getSignCombinations()</a>	121
6.43.3.5	<a href="#">isSpeedOk()</a>	121
6.44	<a href="#">de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter Class Reference</a>	122
6.44.1	<a href="#">Detailed Description</a>	122
6.44.2	<a href="#">Constructor &amp; Destructor Documentation</a>	123
6.44.2.1	<a href="#">TSDSignHistoryAdapter()</a>	123
6.44.3	<a href="#">Member Function Documentation</a>	123
6.44.3.1	<a href="#">getItemCount()</a>	123
6.44.3.2	<a href="#">onBindViewHolder()</a> [1/2]	123
6.44.3.3	<a href="#">onBindViewHolder()</a> [2/2]	124
6.44.3.4	<a href="#">onCreateViewHolder()</a>	124
6.44.3.5	<a href="#">updateTimeDisplay()</a>	125
6.45	<a href="#">de.swp.tsd.trafficsigndetection.TSDSpeedChecker Class Reference</a>	125
6.45.1	<a href="#">Detailed Description</a>	125
6.45.2	<a href="#">Member Function Documentation</a>	125
6.45.2.1	<a href="#">addSigns()</a>	125
6.45.2.2	<a href="#">clear()</a>	126
6.45.2.3	<a href="#">isSpeedOk()</a>	126
6.46	<a href="#">de.swp.tsd.trafficsigndetection.TSDSpeedTracker Class Reference</a>	126
6.46.1	<a href="#">Detailed Description</a>	127
6.46.2	<a href="#">Constructor &amp; Destructor Documentation</a>	127
6.46.2.1	<a href="#">TSDSpeedTracker()</a>	127
6.46.3	<a href="#">Member Function Documentation</a>	128
6.46.3.1	<a href="#">onPause()</a>	128
6.46.3.2	<a href="#">onResume()</a>	128

<b>7</b>	<b>File Documentation</b>	<b>129</b>
7.1	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/MainActivity.java File Reference	129
7.2	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDAboutFragment.java File Reference	129
7.3	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDAutoFocusMode.java File Reference	129
7.4	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDBaseCamera.java File Reference	130
7.5	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraCharacter.java File Reference	130
7.6	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraManager.java File Reference	130
7.7	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraOutputFormat.java File Reference	130
7.8	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCutOffSpeedTracker.java File Reference	131
7.9	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDebugFragment.java File Reference	131
7.10	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDetectorFragment.java File Reference	131
7.11	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDisclaimerFragment.java File Reference	131
7.12	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDisclosureFragment.java File Reference	132
7.13	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDrawable.java File Reference	132
7.14	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDrawableDatabase.java File Reference	132
7.15	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDFragment.java File Reference	132
7.16	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDHelper.java File Reference	133
7.17	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDInitializationErrorDialog.java File Reference	133
7.18	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLensFacing.java File Reference	133
7.19	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLicenseFragment.java File Reference	133
7.20	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLifecycleObserver.java File Reference	134

7.21	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLocationListener.java	File Reference	134
7.22	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDNavigationAdapter.java	File Reference	134
7.23	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDOpticalStabilization↔ Mode.java	File Reference	134
7.24	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPeriodicTask.java	File Reference	135
7.25	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPermissionManager.java	File Reference	135
7.26	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPrivacyFragment.java	File Reference	135
7.27	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDRingMemory.java	File Reference	135
7.28	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDRotationSensor.java	File Reference	136
7.29	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSettingsFragment.java	File Reference	136
7.30	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSetupFragment.java	File Reference	136
7.31	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDShutterCamera.java	File Reference	136
7.32	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDShutterCameraImage↔ Listener.java	File Reference	137
7.33	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSign.java	File Reference	137
7.34	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignCombination.java	File Reference	137
7.35	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignCombinationView↔ Holder.java	File Reference	137
7.36	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignHistory.java	File Reference	138
7.37	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignHistoryAdapter.java	File Reference	138
7.38	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSpeedChecker.java	File Reference	138
7.39	App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSpeedTracker.java	File Reference	138



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">de</a>	9
<a href="#">de.swp</a>	9
<a href="#">de.swp.tsd</a>	9
<a href="#">de.swp.tsd.trafficsigndetection</a>	9





## Chapter 2

# Hierarchical Index

## 2.1 Class Hierarchy

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

Adapter	
de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter . . . . .	122
Callback	
de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback . . . . .	11
de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.FrameEventListener . . . . .	14
de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker . . . . .	45
de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener . . . . .	19
de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.OnCutOffSpeedCalculatedListener . . . . .	20
OnImageAvailableListener	
de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener . . . . .	96
de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionResultListener . . . . .	21
de.swp.tsd.trafficsigndetection.TSDRotationSensor.OnRotationListener . . . . .	23
de.swp.tsd.trafficsigndetection.TSDSpeedTracker.OnSpeedCalculatedListener . . . . .	23
de.swp.tsd.trafficsigndetection.TSDAutoFocusMode . . . . .	25
de.swp.tsd.trafficsigndetection.TSDCameraCharacter . . . . .	31
de.swp.tsd.trafficsigndetection.TSDCameraManager . . . . .	36
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat . . . . .	38
de.swp.tsd.trafficsigndetection.TSDDrawable . . . . .	55
de.swp.tsd.trafficsigndetection.TSDHelper . . . . .	61
de.swp.tsd.trafficsigndetection.TSDLensFacing . . . . .	66
de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode . . . . .	77
de.swp.tsd.trafficsigndetection.TSDPermissionManager . . . . .	80
de.swp.tsd.trafficsigndetection.TSDRingMemory< de.swp.tsd.trafficsigndetection.TSDSignCombination > . . . . .	84
de.swp.tsd.trafficsigndetection.TSDRingMemory< Integer > . . . . .	84
de.swp.tsd.trafficsigndetection.TSDRingMemory< LocationPoint > . . . . .	84
de.swp.tsd.trafficsigndetection.TSDShutterCamera . . . . .	93
de.swp.tsd.trafficsigndetection.TSDSign . . . . .	98
de.swp.tsd.trafficsigndetection.TSDSignCombination . . . . .	110
de.swp.tsd.trafficsigndetection.TSDSpeedChecker . . . . .	125
ViewHolder	
de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder . . . . .	115
ArrayList	
de.swp.tsd.trafficsigndetection.TSDRingMemory< T > . . . . .	84
BaseAdapter	
de.swp.tsd.trafficsigndetection.TSDNavigationAdapter . . . . .	74

DialogFragment	
de.swp.tsd.trafficsigndetection.TSDInitializationErrorDialog . . . . .	65
Fragment	
de.swp.tsd.trafficsigndetection.TSDFragment . . . . .	59
de.swp.tsd.trafficsigndetection.TSDAboutFragment . . . . .	24
de.swp.tsd.trafficsigndetection.TSDDebugFragment . . . . .	47
de.swp.tsd.trafficsigndetection.TSDDetectorFragment . . . . .	49
de.swp.tsd.trafficsigndetection.TSDDisclaimerFragment . . . . .	52
de.swp.tsd.trafficsigndetection.TSDDisclosureFragment . . . . .	53
de.swp.tsd.trafficsigndetection.TSDLicenseFragment . . . . .	68
de.swp.tsd.trafficsigndetection.TSDPrivacyFragment . . . . .	82
de.swp.tsd.trafficsigndetection.TSDSettingsFragment . . . . .	89
de.swp.tsd.trafficsigndetection.TSDSetupFragment . . . . .	91
FragmentActivity	
de.swp.tsd.trafficsigndetection.MainActivity . . . . .	15
LifecycleObserver	
de.swp.tsd.trafficsigndetection.TSDLifecycleObserver . . . . .	69
de.swp.tsd.trafficsigndetection.TSDBaseCamera . . . . .	28
de.swp.tsd.trafficsigndetection.TSDPeriodicTask . . . . .	79
de.swp.tsd.trafficsigndetection.TSDRotationSensor . . . . .	87
de.swp.tsd.trafficsigndetection.TSDSpeedTracker . . . . .	126
LocationListener	
de.swp.tsd.trafficsigndetection.TSDLocationListener . . . . .	72
ViewModel	
de.swp.tsd.trafficsigndetection.TSDDrawableDatabase . . . . .	58
de.swp.tsd.trafficsigndetection.TSDSignHistory . . . . .	119

## Chapter 3

# Class Index

### 3.1 Class List

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

de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback	11
de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.FrameEventListener	14
de.swp.tsd.trafficsigndetection.MainActivity	15
de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener	19
de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.OnCutOffSpeedCalculatedListener	20
de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionResultListener	21
de.swp.tsd.trafficsigndetection.TSDRotationSensor.OnRotationListener	23
de.swp.tsd.trafficsigndetection.TSDSpeedTracker.OnSpeedCalculatedListener	23
de.swp.tsd.trafficsigndetection.TSDAboutFragment	24
de.swp.tsd.trafficsigndetection.TSDAutoFocusMode	25
de.swp.tsd.trafficsigndetection.TSDBaseCamera	28
de.swp.tsd.trafficsigndetection.TSDCameraCharacter	31
de.swp.tsd.trafficsigndetection.TSDCameraManager	36
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat	38
de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker	45
de.swp.tsd.trafficsigndetection.TSDDebugFragment	47
de.swp.tsd.trafficsigndetection.TSDDetectorFragment	49
de.swp.tsd.trafficsigndetection.TSDDisclaimerFragment	52
de.swp.tsd.trafficsigndetection.TSDDisclosureFragment	53
de.swp.tsd.trafficsigndetection.TSDDrawable	55
de.swp.tsd.trafficsigndetection.TSDDrawableDatabase	58
de.swp.tsd.trafficsigndetection.TSDFragment	59
de.swp.tsd.trafficsigndetection.TSDHelper	61
de.swp.tsd.trafficsigndetection.TSDInitializationErrorDialog	65
de.swp.tsd.trafficsigndetection.TSDLensFacing	66
de.swp.tsd.trafficsigndetection.TSDLicenseFragment	68
de.swp.tsd.trafficsigndetection.TSDLifecycleObserver	69
de.swp.tsd.trafficsigndetection.TSDLocationListener	72
de.swp.tsd.trafficsigndetection.TSDNavigationAdapter	74
de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode	77
de.swp.tsd.trafficsigndetection.TSDPeriodicTask	79
de.swp.tsd.trafficsigndetection.TSDPermissionManager	80
de.swp.tsd.trafficsigndetection.TSDPrivacyFragment	82
de.swp.tsd.trafficsigndetection.TSDRingMemory< T >	84
de.swp.tsd.trafficsigndetection.TSDRotationSensor	87

de.swp.tsd.trafficsigndetection.TSDSettingsFragment	89
de.swp.tsd.trafficsigndetection.TSDSetupFragment	91
de.swp.tsd.trafficsigndetection.TSDShutterCamera	93
de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener	96
de.swp.tsd.trafficsigndetection.TSDSign	98
de.swp.tsd.trafficsigndetection.TSDSignCombination	110
de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder	115
de.swp.tsd.trafficsigndetection.TSDSignHistory	119
de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter	122
de.swp.tsd.trafficsigndetection.TSDSpeedChecker	125
de.swp.tsd.trafficsigndetection.TSDSpeedTracker	126

## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/MainActivity.java	129
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDAboutFragment.java	129
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDAutoFocusMode.java	129
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDBaseCamera.java	130
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraCharacter.java	130
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraManager.java	130
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraOutputFormat.java	130
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCutOffSpeedTracker.java	131
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDebugFragment.java	131
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDetectorFragment.java	131
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDisclaimerFragment.java	131
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDisclosureFragment.java	132
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDrawable.java	132
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDrawableDatabase.java	132
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDFragment.java	132
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDHelper.java	133
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDInitializationErrorDialog.java	133
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLensFacing.java	133
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLicenseFragment.java	133
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLifecycleObserver.java	134
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLocationListener.java	134
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDNavigationAdapter.java	134
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDOpticalStabilizationMode.java	134
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPeriodicTask.java	135
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPermissionManager.java	135
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPrivacyFragment.java	135
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDRingMemory.java	135
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDRotationSensor.java	136
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSettingsFragment.java	136
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSetupFragment.java	136
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDShutterCamera.java	136
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDShutterCameraImageListener.java	137
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSign.java	137
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignCombination.java	137

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TDSignCombinationViewHolder. <a href="#">↵</a>	
<a href="#">java</a> . . . . .	137
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TDSignHistory.java . . . . .	138
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TDSignHistoryAdapter.java . . . . .	138
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TDSpeedChecker.java . . . . .	138
App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TDSpeedTracker.java . . . . .	138

## Chapter 5

# Namespace Documentation

### 5.1 Package de

#### Packages

- package [swp](#)

### 5.2 Package de.swp

#### Packages

- package [tsd](#)

### 5.3 Package de.swp.tsd

#### Packages

- package [trafficsigndetection](#)

### 5.4 Package de.swp.tsd.trafficsigndetection

#### Classes

- class [MainActivity](#)
- class [TSDAboutFragment](#)
- enum [TSDAutoFocusMode](#)
- class [TSDBaseCamera](#)
- class [TSDCameraCharacter](#)
- class [TSDCameraManager](#)
- enum [TSDCameraOutputFormat](#)
- class [TSDCutOffSpeedTracker](#)

- class [TSDDDebugFragment](#)
- class [TSDDetectorFragment](#)
- class [TSDDisclaimerFragment](#)
- class [TSDDisclosureFragment](#)
- class [TSDDrawable](#)
- class [TSDDrawableDatabase](#)
- class [TSDFragment](#)
- class [TSDHelper](#)
- class [TSDInitializationErrorDialog](#)
- enum [TSDLensFacing](#)
- class [TSDLicenseFragment](#)
- class [TSDLifecycleObserver](#)
- class [TSDLocationListener](#)
- class [TSDNavigationAdapter](#)
- enum [TSDOpticalStabilizationMode](#)
- class [TSDPeriodicTask](#)
- class [TSDPermissionManager](#)
- class [TSDPrivacyFragment](#)
- class [TSDRingMemory](#)
- class [TSDRotationSensor](#)
- class [TSDSettingsFragment](#)
- class [TSDSetupFragment](#)
- class [TSDShutterCamera](#)
- class [TSDShutterCameraImageListener](#)
- enum [TSDSign](#)
- class [TSDSignCombination](#)
- class [TSDSignCombinationViewHolder](#)
- class [TSDSignHistory](#)
- class [TSDSignHistoryAdapter](#)
- class [TSDSpeedChecker](#)
- class [TSDSpeedTracker](#)

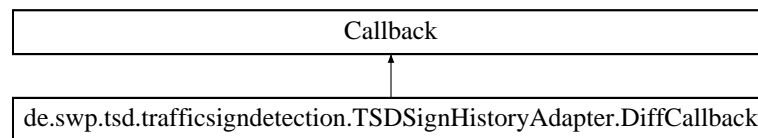


## Chapter 6

# Class Documentation

### 6.1 de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback:



#### Public Member Functions

- int [getOldListSize](#) ()
- int [getNewListSize](#) ()
- boolean [areItemsTheSame](#) (int oldPos, int newPos)
- boolean [areContentsTheSame](#) (int oldPos, int newPos)
- Object [getChangePayload](#) (int oldPos, int newPos)

#### Package Functions

- [DiffCallback](#) ([TSDRingMemory](#)< [TSDSignCombination](#) > oldMemory, [TSDRingMemory](#)< [TSDSignCombination](#) > newMemory)

#### 6.1.1 Detailed Description

This class defines a callback for the DiffUtils to find the difference between two TSDRingMemories with TSDSignCombinations. This is used to identify the specific changes when the life data of the sign history signals an update (Since it does not tell what changed in the list).

#### 6.1.2 Constructor & Destructor Documentation

### 6.1.2.1 DiffCallback()

```
de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback.DiffCallback (
    TSDRingMemory< TSDSignCombination > oldMemory,
    TSDRingMemory< TSDSignCombination > newMemory ) [inline], [package]
```

This constructor creates a new instance of [DiffCallback](#) to be used to find differences between two TSDRingMemories with TSDSignCombinations.

#### Parameters

<i>oldMemory</i>	Old collection.
<i>newMemory</i>	New Collection.

## 6.1.3 Member Function Documentation

### 6.1.3.1 areContentsTheSame()

```
boolean de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback.areContentsTheSame (
    int oldPos,
    int newPos ) [inline]
```

This method checks, when the sign were told to be the same by [DiffCallback#areItemsTheSame\(int, int\)](#), whether the content was updated or not.

#### Parameters

<i>oldPos</i>	Index of the combi in the old list.
<i>newPos</i>	Index of the combi in the new list.

#### Returns

Whether the content of the combi was updated or not.

### 6.1.3.2 areItemsTheSame()

```
boolean de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback.areItemsTheSame (
    int oldPos,
    int newPos ) [inline]
```

This method checks whether some sign combinations have the same ids.

**Parameters**

<i>oldPos</i>	Index of the combi in the old list.
<i>newPos</i>	Index of the combi in the new list.

**Returns**

Whether the combis are the same or not.

**6.1.3.3 getChangePayload()**

```
Object de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback.getChangePayload (
    int oldPos,
    int newPos ) [inline]
```

This method just returns the old sign combi as payload. The differences are found out in TSDSignHistoryAdapter#onBindViewHolder(TSDSignCombinationViewHolder, int, List).

**Parameters**

<i>oldPos</i>	Index of the combi in the old list.
<i>newPos</i>	Index of the combi in the new list.

**Returns**

Old sign combination.

**6.1.3.4 getNewListSize()**

```
int de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback.getNewListSize ( )
[inline]
```

This method queries the size of the new list.

**Returns**

Size of the new list.

### 6.1.3.5 getOldListSize()

```
int de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback.getOldListSize ( )
[inline]
```

This method queries the size of the old list.

#### Returns

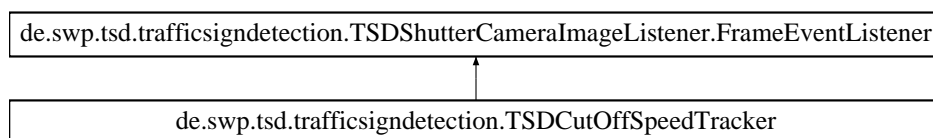
Size of the old list.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDSignHistoryAdapter.java](#)

## 6.2 de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.FrameEvent↔ Listener Interface Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.FrameEventListener:



### Public Member Functions

- void [onFrameFinished](#) (long duration)

### 6.2.1 Detailed Description

This interface is used to receive events of the frame processing. Actually it is only for the event when frames are finished to get processed.

### 6.2.2 Member Function Documentation

#### 6.2.2.1 onFrameFinished()

```
void de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.FrameEvent↔
Listener.onFrameFinished (
    long duration )
```

This method is called every time one camera frame is finished.

## Parameters

<i>duration</i>	Time it took to process the frame in milliseconds.
-----------------	--

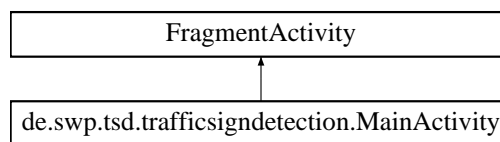
Implemented in [de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker](#).

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDShutterCameraImageListener.java

## 6.3 de.swp.tsd.trafficsigndetection.MainActivity Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.MainActivity:



### Public Member Functions

- void [onBackPressed](#) ()
- void [onRequestPermissionsResult](#) (int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults)
- CoordinatorLayout [getContentLayout](#) ()
- void [onClickMenu](#) (View view)
- void [onPrivacyPolicyAccepted](#) (View view)
- void [onDisclosureAccepted](#) (View view)
- void [onDisclaimerAccepted](#) (View view)
- void [openDetectorFragment](#) (View view)

### Protected Member Functions

- void [onCreate](#) (Bundle savedInstanceState)

### Static Package Functions

- [\[static initializer\]](#)



### 6.3.2.3 onBackPressed()

```
void de.swp.tsd.trafficsigndetection.MainActivity.onBackPressed ( ) [inline]
```

This method is called every time the back button is pressed. It is used to run the standard action. Afterwards it updates the title shown in the app.

References `de.swp.tsd.trafficsigndetection.TSDFragment.displayNameId`.

### 6.3.2.4 onClickMenu()

```
void de.swp.tsd.trafficsigndetection.MainActivity.onClickMenu (
    View view ) [inline]
```

This method gets called from UI and opens the hamburger menu.

#### Parameters

<i>view</i>	View the method is called from.
-------------	---------------------------------

### 6.3.2.5 onCreate()

```
void de.swp.tsd.trafficsigndetection.MainActivity.onCreate (
    Bundle savedInstanceState ) [inline], [protected]
```

This method is called on the startup of this activity. It only loads some basic [fragments](#) at start until the privacy policy, the disclosure and the disclaimer gets accepted and it initializes the native RoadSignAPI.

#### Parameters

<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last stop of the app. Currently this does not get used.
---------------------------	--

References `de.swp.tsd.trafficsigndetection.TSDNavigationAdapter.getItem()`, `de.swp.tsd.trafficsigndetection.MainActivity.onPrivacyPolicyAccepted()`, and `de.swp.tsd.trafficsigndetection.TSDHelper.readBooleanFromSharedPreferences()`.

### 6.3.2.6 onDisclaimerAccepted()

```
void de.swp.tsd.trafficsigndetection.MainActivity.onDisclaimerAccepted (
    View view ) [inline]
```

This method gets called from UI when the disclaimer was accepted and loads all relevant [fragments](#) for the app usage. It also shows an error dialog when the initialization of the api failed.

## Parameters

<i>view</i>	View the method is called from.
-------------	---------------------------------

## 6.3.2.7 onDisclosureAccepted()

```
void de.swp.tsd.trafficsigndetection.MainActivity.onDisclosureAccepted (
    View view ) [inline]
```

This method gets called from UI when the disclosure was accepted and loads all relevant [fragments](#) for the disclaimer. It also saves a flag to the shared preferences indicating that the user accepted the disclosure. So at the next start it does not has to be shown again.

## Parameters

<i>view</i>	View the method is called from.
-------------	---------------------------------

References `de.swp.tsd.trafficsigndetection.TSDHelper.writeBooleanToSharedPreferences()`.

## 6.3.2.8 onPrivacyPolicyAccepted()

```
void de.swp.tsd.trafficsigndetection.MainActivity.onPrivacyPolicyAccepted (
    View view ) [inline]
```

This method gets called from UI when the privacy policy was accepted and loads all relevant [fragments](#) for the disclosure. It also saves a flag to the shared preferences indicating that the user accepted the privacy policy. So at the next start it does not has to be shown again. The disclosure gets shown, when the camera permission was not granted, the location permission not granted and the speed tracking is enabled, or when the disclosure was not shown at least once, which is checked by a flag in the shared preferences. In short: It gets shown when teh user is likely to come across permission requests during his following usage of the app.

## Parameters

<i>view</i>	View the method is called from.
-------------	---------------------------------

References `de.swp.tsd.trafficsigndetection.MainActivity.onDisclosureAccepted()`, `de.swp.tsd.trafficsigndetection.TSDHelper.readBooleanFromSharedPreferences()`, and `de.swp.tsd.trafficsigndetection.TSDHelper.writeBooleanToSharedPreferences()`.

## 6.3.2.9 onRequestPermissionsResult()

```
void de.swp.tsd.trafficsigndetection.MainActivity.onRequestPermissionsResult (
    int requestCode,
```



```
@NonNull String [] permissions,
@NonNull int [] grantResults ) [inline]
```

This method is called every time a permission request result gets processed. It forwards the delivered result information to Activity, int[], CoordinatorLayout) onRequestPermissionsResult.

#### Parameters

<i>requestCode</i>	The code set when the permission was requested. This can be used to identify the result.
<i>permissions</i>	Array of strings containing the requested permissions.
<i>grantResults</i>	Array containing the users choice for every requested permission.

References de.swp.tsd.trafficsigndetection.TSDPermissionManager.onRequestPermissionsResult().

#### 6.3.2.10 openDetectorFragment()

```
void de.swp.tsd.trafficsigndetection.MainActivity.openDetectorFragment (
    View view ) [inline]
```

This method gets called from UI and opens the detector fragment. This is used from [TSDSetupFragment](#) its user interface to finish setup.

#### Parameters

<i>view</i>	View the method is called from.
-------------	---------------------------------

References de.swp.tsd.trafficsigndetection.TSDFragment.displayNameId, de.swp.tsd.trafficsigndetection.TSDNavigationAdapter.getCount(), and de.swp.tsd.trafficsigndetection.TSDNavigationAdapter.getItem().

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[MainActivity.java](#)

## 6.4 de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener Interface Reference

### Public Member Functions

- void [onResume](#) ()
- void [onOpened](#) (CameraDevice camera)
- void [onClose](#) ()
- void [onPaused](#) ()

#### 6.4.1 Detailed Description

This interface defines an observer to be implemented by users of the [TSDBaseCamera](#) class signaling them the events in the lifecycle of the camera.

## 6.4.2 Member Function Documentation

### 6.4.2.1 onClose()

```
void de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener.onClose ( )
```

This method gets called before the camera gets closed. It should be used for destroying saved session objects.

### 6.4.2.2 onOpened()

```
void de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener.onOpened (
    CameraDevice camera )
```

This method gets called when the camera was successfully opened. It can be used to start a capture session.

#### Parameters

<i>camera</i>	Camera device having been opened.
---------------	-----------------------------------

### 6.4.2.3 onPauseed()

```
void de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener.onPauseed ( )
```

This method gets called after the camera was closed. It should be used for closing initialized image readers.

### 6.4.2.4 onResume()

```
void de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener.onResume ( )
```

This method gets called when the camera should get started or resumed. It should be used for setting up targets. When having this done, the camera resume can be continued by calling [onTargetsReady\(\)](#).

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDBaseCamera.java](#)

## 6.5 de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.OnCutOffSpeedCalculatedListener Interface Reference ↩

### Public Member Functions

- void [onCutOffSpeedCalculated](#) (Integer cutOffSpeed)

### 6.5.1 Detailed Description

This interface defines an observer to be implemented by users of the [TSDCutOffSpeedTracker](#) signaling them when a new cut-off speed was calculated.

### 6.5.2 Member Function Documentation

#### 6.5.2.1 onCutOffSpeedCalculated()

```
void de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.OnCutOffSpeedCalculatedListener.↔  
onCutOffSpeedCalculated (   
    Integer cutOffSpeed )
```

This method gets called whenever a new cut-off speed was calculated. The cut-off speed describes the maximum speed the device can move while still being fast enough to detect signs.

#### Parameters

<i>cutOffSpeed</i>	Maximum speed in km/h which still allows the device to detect the signs on the captured images. If this is null, then no cut-off speed can currently be given.
--------------------	--

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCutOffSpeedTracker.java](#)

## 6.6 de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionResultListener Interface Reference↔

### Public Member Functions

- String [] [getRequestedPermissions](#) ()
- void [onRequestedPermissionsGranted](#) ()
- void [onRequestedPermissionsDeclined](#) ()
- String [getDeclineText](#) ()

### 6.6.1 Detailed Description

This interface defines an observer to be implemented by all users wanting to request some permission signaling them, when the users decision is known.

#### See also

[TSDPermissionManager](#)

## 6.6.2 Member Function Documentation

### 6.6.2.1 getDeclineText()

```
String de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionResultListener.↵  
getDeclineText ( )
```

This method is called when some of the requested permissions were not granted. It should return a message that can be showed to the user.

#### Returns

Decline message text

### 6.6.2.2 getRequestedPermissions()

```
String [ ] de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionResult↵  
Listener.getRequestedPermissions ( )
```

This method should return the permissions this request asks for.

#### Returns

String array of requested permissions

### 6.6.2.3 onRequestedPermissionsDeclined()

```
void de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionResultListener.↵  
onRequestedPermissionsDeclined ( )
```

This method gets called, when the user decided to decline at least one of the requested permissions.

### 6.6.2.4 onRequestedPermissionsGranted()

```
void de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionResultListener.↵  
onRequestedPermissionsGranted ( )
```

This method gets called, when the users decided to grant all requested permissions.

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPermissionManager.java](#)

## 6.7 de.swp.tsd.trafficsigndetection.TSDRotationSensor.OnRotationListener Interface Reference

### Public Member Functions

- void [onRotation](#) (int roll, int pitch)

#### 6.7.1 Detailed Description

This interface defines the listener interface for rotation events. Implement this and give it to a [TSDRotationSensor](#) constructor to get the rotation of the device every time it changes.

#### 6.7.2 Member Function Documentation

##### 6.7.2.1 onRotation()

```
void de.swp.tsd.trafficsigndetection.TSDRotationSensor.OnRotationListener.onRotation (
    int roll,
    int pitch )
```

This method gets called on every sensor event delivering the roll and pitch of the device.

##### Parameters

<i>roll</i>	Device roll (z-axis rotation) in degrees.
<i>pitch</i>	Device pitch (y-axis rotation) in degrees.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDRotationSensor.java](#)

## 6.8 de.swp.tsd.trafficsigndetection.TSDSpeedTracker.OnSpeedCalculatedListener Interface Reference

### Public Member Functions

- void [onSpeedCalculated](#) (Integer speed)

#### 6.8.1 Detailed Description

This interface defines an observer to be implemented by users of the [TSDSpeedTracker](#) class signaling them when the speed was calculated.

## 6.8.2 Member Function Documentation

### 6.8.2.1 onSpeedCalculated()

```
void de.swp.tsd.trafficsigndetection.TSDSpeedTracker.OnSpeedCalculatedListener.onSpeedCalculated
(
    Integer speed )
```

This method gets called every time the speed was calculated.

#### Parameters

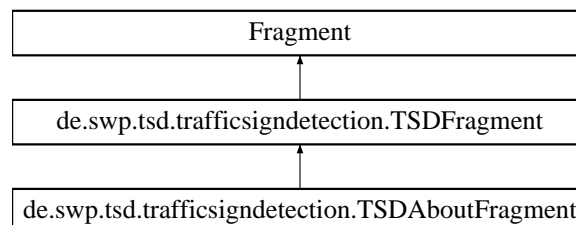
<i>speed</i>	Speed of the device in kilometers per hour. This can be null, if the speed is disabled or negative, if it is currently not available.
--------------	---

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSpeedTracker.java](#)

## 6.9 de.swp.tsd.trafficsigndetection.TSDAboutFragment Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDAboutFragment:



### Public Member Functions

- [TSDAboutFragment \(\)](#)
- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

### Additional Inherited Members

#### 6.9.1 Detailed Description

This [TSDFragment](#) displays the user some information about the app. It is the point where the user can read the privacy policy and the disclaimer again.

## 6.9.2 Constructor & Destructor Documentation

### 6.9.2.1 TSDAboutFragment()

```
de.swp.tsd.trafficsigndetection.TSDAboutFragment.TSDAboutFragment ( ) [inline]
```

This constructor creates a new instance of [TSDAboutFragment](#) setting the correct values for [isRelevant name](#) and [isRelevant icon](#).

## 6.9.3 Member Function Documentation

### 6.9.3.1 onCreateView()

```
View de.swp.tsd.trafficsigndetection.TSDAboutFragment.onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState ) [inline]
```

This method gets called every time the view of the [TSDFragment](#) is newly created. It inflates the fragment\_about layout as user interface for the [TSDFragment](#).

See also

[TSDAboutFragment](#)

#### Parameters

<i>inflater</i>	Inflater to use for inflating the user interface layout of this fragment.
<i>container</i>	The place the ui should be inflated to, but not attached.
<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.

#### Returns

The inflated view containing the user interface of this [TSDFragment](#).

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDAboutFragment.java](#)

## 6.10 de.swp.tsd.trafficsigndetection.TSDAutoFocusMode Enum Reference

### Public Member Functions

- [TSDAutoFocusMode](#) (int [value](#))

## Static Public Member Functions

- static [TSDAutoFocusMode fromInt](#) (int [value](#))

## Public Attributes

- [OFF](#) =(0)
- [AUTO](#) =(1)
- [MACRO](#) =(2)
- [CONTINUOUS\\_VIDEO](#) =(3)
- [CONTINUOUS\\_PICTURE](#) =(4)
- [EDOF](#) =(5)
- final int [value](#)

### 6.10.1 Detailed Description

This enum wraps the possible autofocus modes of the camera instance according to the [Android API](#) into an enum. See the documentation for detailed information.

### 6.10.2 Constructor & Destructor Documentation

#### 6.10.2.1 TSDAutoFocusMode()

```
de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.TSDAutoFocusMode (
    int value ) [inline]
```

This constructor creates a new enum instance setting the internal enum value to the given one.

#### Parameters

<i>value</i>	This value should match the value of the constant in the Android API describing the same mode.
--------------	--

### 6.10.3 Member Function Documentation

#### 6.10.3.1 fromInt()

```
static TSDAutoFocusMode de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.fromInt (
    int value ) [inline], [static]
```

This function can convert a given value which is one of the constant describing autofocus modes in the Android API to the corresponding enum value. If the given value does not match any defined enum, then OFF will be returned.



#### Parameters

<i>value</i>	Integer that should be converted.
--------------	-----------------------------------

#### Returns

Enum value that corresponds to the given int value. If not matching one is found, OFF will be returned.

### 6.10.4 Member Data Documentation

#### 6.10.4.1 AUTO

```
de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.AUTO = (1)
```

This will activate the autofocus jumping in when it gets triggered. `CONTROL_AF_MODE_AUTO`

#### 6.10.4.2 CONTINUOUS\_PICTURE

```
de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.CONTINUOUS_PICTURE = (4)
```

This will activate the autofocus for continuous picture taking jumping in on trigger and trying to focus as fast as possible. `CONTROL_AF_MODE_CONTINUOUS_PICTURE`

#### 6.10.4.3 CONTINUOUS\_VIDEO

```
de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.CONTINUOUS_VIDEO = (3)
```

This will activate the autofocus for videos trying to achieve a constant focused image stream. `CONTROL_AF_MODE_CONTINUOUS_VIDEO`

#### 6.10.4.4 EDOF

```
de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.EDOF = (5)
```

This will activate an autofocus mode automatically delivering extended depth of field effects. `CONTROL_AF_MODE_EDOF`

#### 6.10.4.5 MACRO

```
de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.MACRO = (2)
```

This will activate the autofocus for close-up captures when it gets triggered. `CONTROL_AF_MODE_MACRO`

#### 6.10.4.6 OFF

```
de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.OFF = (0)
```

The autofocus is disabled and lens distance gets controlled by the application. [CONTROL\\_AF\\_MODE\\_OFF](#)

#### 6.10.4.7 value

```
final int de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.value
```

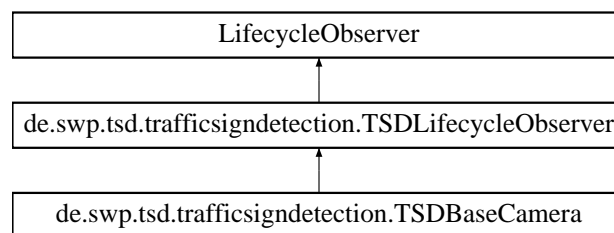
This value describes the value of the constant in the Android API describing the same mode this enum wraps.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDAutoFocusMode.java](#)

## 6.11 de.swp.tsd.trafficsigndetection.TSDBaseCamera Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDBaseCamera:



### Classes

- interface [OnCameraEventListener](#)

### Public Member Functions

- [TSDCameraCharacter](#) [getCharacter](#) ()
- Handler [getBackgroundHandler](#) ()
- CameraDevice.StateCallback [getCameraStateCallback](#) ()
- void [onTargetsReady](#) ()

### Protected Member Functions

- void [onResume](#) ()
- void [onPause](#) ()

## Package Functions

- [TSDBaseCamera](#) (@NonNull [TSDCameraCharacter](#) character, @NonNull [OnCameraEventListener](#) eventListener, @NonNull Lifecycle lifecycle)

### 6.11.1 Detailed Description

This class is the base for every specialized camera used in this application. Its task is to manage the opening and closing of the given CameraDevice taking the lifecycle of the parent into consideration closing and reopening the camera when the parent gets paused and resumed. Using the [OnCameraEventListener](#) the behaviour of the camera can be defined. The lifecycle handling is done by the [TSDLifecycleObserver](#) base class. The camera is run inside a background thread which gets started and stopped together with the camera itself. Startup and Stopping are locked to each other by a semaphore avoiding that the camera could be stopped before it was even completely started. Please mind, that this class does not handle the runtime camera permission. This needs to be done outside the class. Otherwise the camera initialization will fail. To define a new camera behaviour based on this class you need to create an [OnCameraEventListener](#). This listener gets signaled on every important camera event and should be used for defining the behavior and the used camera target surfaces.

See also

[OnCameraEventListener](#)

### 6.11.2 Constructor & Destructor Documentation

#### 6.11.2.1 TSDBaseCamera()

```
de.swp.tsd.trafficsigndetection.TSDBaseCamera.TSDBaseCamera (
    @NonNull TSDCameraCharacter character,
    @NonNull OnCameraEventListener eventListener,
    @NonNull Lifecycle lifecycle ) [inline], [package]
```

This constructor creates a new instance of [TSDBaseCamera](#) setting the listener on the parents lifecycle. This does not enable the camera. To enable the camera please see [TSDLifecycleObserver#enable\(\)](#).

#### Parameters

<i>character</i>	Character defining the camera to use. This can be obtained by using the <a href="#">TSDCameraManager</a> .
<i>eventListener</i>	Observer for all important camera events. This should be used to define the behaviour of the camera itself.
<i>lifecycle</i>	Lifecycle of the parent starting and stopping the camera and background thread.

### 6.11.3 Member Function Documentation

### 6.11.3.1 `getBackgroundHandler()`

```
Handler de.swp.tsd.trafficsigndetection.TSDBaseCamera.getBackgroundHandler ( ) [inline]
```

Getter for the loop handler of the background thread for doing the camera work. You can use this handler to let work get done on this thread. Especially use it for camera related work inside the [OnCameraEventListener](#).

#### Returns

Handler of the background thread loop

### 6.11.3.2 `getCameraStateCallback()`

```
CameraDevice.StateCallback de.swp.tsd.trafficsigndetection.TSDBaseCamera.getCameraState↵  
Callback ( ) [inline]
```

Getter for the state callback object of the camera. This callback is used by the camera api to signal back the state of the camera. The implementation releases the in the startup locked semaphore and signals the [OnCameraEvent↵  
Listener](#) with the correct signal according to the state of the camera.

#### Returns

State callback object of the camera

### 6.11.3.3 `getCharacter()`

```
TSDCameraCharacter de.swp.tsd.trafficsigndetection.TSDBaseCamera.getCharacter ( ) [inline]
```

Getter for the character of the camera used. This is the same that was given to the constructor.

#### Returns

Character of the camera being used

### 6.11.3.4 `onPause()`

```
void de.swp.tsd.trafficsigndetection.TSDBaseCamera.onPause ( ) [inline], [protected]
```

This method gets called when the parent lifecycle hits the ON\_PAUSE event and the object is enabled. It closes the camera, stops the background thread and signals [OnCameraEventListener#onPaused\(\)](#).

References [de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getManager\(\)](#), [de.swp.tsd.trafficsigndetection.↵  
TSDBaseCamera.OnCameraEventListener.onClose\(\)](#), [de.swp.tsd.trafficsigndetection.TSDBaseCamera.On↵  
CameraEventListener.onPaused\(\)](#), and [de.swp.tsd.trafficsigndetection.TSDCameraManager.openCamera\(\)](#).

## 6.11.3.5 onResume()

```
void de.swp.tsd.trafficsigndetection.TSDBaseCamera.onResume ( ) [inline], [protected]
```

This method gets called when the parent lifecycle hits the ON\_RESUME event and the object is enabled. It starts the background thread and signals [OnCameraEventListener#onResume\(\)](#).

References [de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener.onResume\(\)](#).

## 6.11.3.6 onTargetsReady()

```
void de.swp.tsd.trafficsigndetection.TSDBaseCamera.onTargetsReady ( ) [inline]
```

This method should only be called after having received the [onResume\(\)](#) callback of the [OnCameraEventListener](#) and having initialized all necessary camera targets.

See also

[OnCameraEventListener::onResume\(\)](#)

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDBaseCamera.java](#)

## 6.12 de.swp.tsd.trafficsigndetection.TSDCameraCharacter Class Reference

## Public Member Functions

- String [getId](#) ()
- [TSDCameraManager](#) [getManager](#) ()
- CameraCharacteristics [getCameraCharacteristics](#) ()
- Integer [getSensorRotation](#) ()
- [TSDLensFacing](#) [getLensFacing](#) ()
- [TSDAutoFocusMode](#) [] [getAvailableAutoFocusModes](#) ()
- [TSDOpticalStabilizationMode](#) [] [getAvailableOisModes](#) ()
- float [] [getAvailableApertures](#) ()
- [TSDCameraOutputFormat](#) [] [getAvailableOutputFormats](#) ()
- Size [] [getAvailableOutputSizes](#) ([TSDCameraOutputFormat](#) format)
- Double [getMaximumViewAngle](#) ()
- Size [getMatchingOutputSize](#) ([TSDCameraOutputFormat](#) format, Size targetSize)
- int [getScore](#) ([TSDLensFacing](#) facing)

## Package Functions

- [TSDCameraCharacter](#) (String id, CameraCharacteristics character, [TSDCameraManager](#) manager)

### 6.12.1 Detailed Description

Describes the characteristics or features of a corresponding camera instance. This is an advanced wrapper around `CameraCharacteristics` implementing methods to get certain information when necessary already converted from constants to wrapper enums like [TSDAutoFocusMode](#).

### 6.12.2 Constructor & Destructor Documentation

#### 6.12.2.1 TSDCameraCharacter()

```
de.swp.tsd.trafficsigndetection.TSDCameraCharacter.TSDCameraCharacter (
    String id,
    CameraCharacteristics character,
    TSDCameraManager manager ) [inline], [package]
```

This constructor creates a new instance wrapping the given information for later use.

#### Parameters

<i>id</i>	Id of the corresponding camera character queried by the given camera manager.
<i>character</i>	Characteristics of the corresponding camera queried by the given camera manager.
<i>manager</i>	Camera manager the given information were queried with.

### 6.12.3 Member Function Documentation

#### 6.12.3.1 getAvailableApertures()

```
float [] de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getAvailableApertures ( ) [inline]
```

This method queries all of the available apertures for the corresponding camera instance.

#### Returns

Array of available apertures. In case there are no such information, this will be empty.

### 6.12.3.2 getAvailableAutoFocusModes()

```
TSDAutoFocusMode [] de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getAvailableAutoFocusModes ( ) [inline]
```

This method queries all of the available modes for autofocus of the corresponding camera instance.

#### Returns

Array of autofocus modes. In case there are no such information, this will be empty.

References de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.fromInt().

### 6.12.3.3 getAvailableOisModes()

```
TSDOpticalStabilizationMode [] de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getAvailableOisModes ( ) [inline]
```

This method queries all of the available modes for optical image stabilization of the corresponding camera instance.

#### Returns

Array of ois modes. In case there are no such information, this will be empty.

References de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode.fromInt().

### 6.12.3.4 getAvailableOutputFormats()

```
TSDCameraOutputFormat [] de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getAvailableOutputFormats ( ) [inline]
```

This method queries all of the available output formats for the corresponding camera instance.

#### Returns

Array of available output formats. This array can be empty.

References de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.fromInt().

### 6.12.3.5 getAvailableOutputSizes()

```
Size [] de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getAvailableOutputSizes ( TSDCameraOutputFormat format ) [inline]
```

This method queries the available output image sizes of the corresponding camera instance for the given image format.

**Parameters**

<i>format</i>	The format available sizes should be queried for.
---------------	---

**Returns**

Array of available output sizes for the format.

References `de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.value`.

**6.12.3.6 getCameraCharacteristics()**

```
CameraCharacteristics de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getCameraCharacteristics
( ) [inline]
```

Getter for raw characteristics of this camera. Use the already given methods to query features instead if sufficient. This is the character given the instance at construction.

**Returns**

Raw camera characteristics

**6.12.3.7 getId()**

```
String de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getId ( ) [inline]
```

Getter for the camera id. This is an hardcoded value unique to every camera device inside the device used to identify it.

**Returns**

Camera id

**6.12.3.8 getLensFacing()**

```
TSDLensFacing de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getLensFacing ( ) [inline]
```

This method queries the facing direction of the corresponding camera.

**Returns**

The facing of the camera. In case there is no such information available, `EXTERNAL` will be returned.

References `de.swp.tsd.trafficsigndetection.TSDLensFacing.EXTERNAL`, and `de.swp.tsd.trafficsigndetection.TSD↔LensFacing.fromInt()`.



## 6.12.3.9 getManager()

```
TSDCameraManager de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getManager ( ) [inline]
```

Getter for the parent camera manager used to query these characteristics.

## Returns

Camera manager

## 6.12.3.10 getMatchingOutputSize()

```
Size de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getMatchingOutputSize (
    TSDCameraOutputFormat format,
    Size targetSize ) [inline]
```

This method queries for the best matching at least as big supported output image size of this camera for given target size, aspect ration and format.

## Parameters

<i>format</i>	Format of the output image of the camera which should have this queried size. This needs to be a by the camera supported format and can be queried by <a href="#">TSDCameraCharacter#getAvailableOutputFormats()</a> .
<i>targetSize</i>	Target size for the image the camera should output. The returned size will be at least as big as this target size and will have the same aspect ratio.

## Returns

Either the found supported size or null, if no matching size was found.

References [de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getAvailableOutputSizes\(\)](#).

## 6.12.3.11 getMaximumViewAngle()

```
Double de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getMaximumViewAngle ( ) [inline]
```

This method calculates the maximum supported view angle of the corresponding camera in radians. For more information at the calculation take a look at [this question](#) on stackoverflow. Internally it calculates both the vertical and horizontal view angle and returns the greater one of them.

## Returns

Either the maximum supported view angle in radians or null, if something went wrong.

### 6.12.3.12 `getScore()`

```
int de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getScore (
    TSDLensFacing facing ) [inline]
```

This method calculates a score for this camera character. This can be used to compare the cameras built into an device and to choose the best one. A bigger score usually means a better camera for our use case in this app. The score is calculated using the direction of the camera as biggest influence (it should match the given direction). Furthermore the number of available Ois, the minimum supported aperture (smaller is better) and the availability of the autofocus modes AUTO and OFF are taken into consideration for a better score.

#### Parameters

<i>facing</i>	The direction the camera should face
---------------	--------------------------------------

#### Returns

Score of this camera

References `de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.AUTO`, `de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getAvailableApertures()`, `de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getAvailableAutoFocusModes()`, `de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getAvailableOisModes()`, `de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getLensFacing()`, and `de.swp.tsd.trafficsigndetection.TSDAutoFocusMode.OFF`.

### 6.12.3.13 `getSensorRotation()`

```
Integer de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getSensorRotation ( ) [inline]
```

Queries the clockwise angle the output image needs to be rotated to be upright on the screen of the device in its native orientation.

#### Returns

Sensor orientation in [0, 90, 180, 270] degrees. If the value could not be queried, then null will be returned.

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraCharacter.java](#)

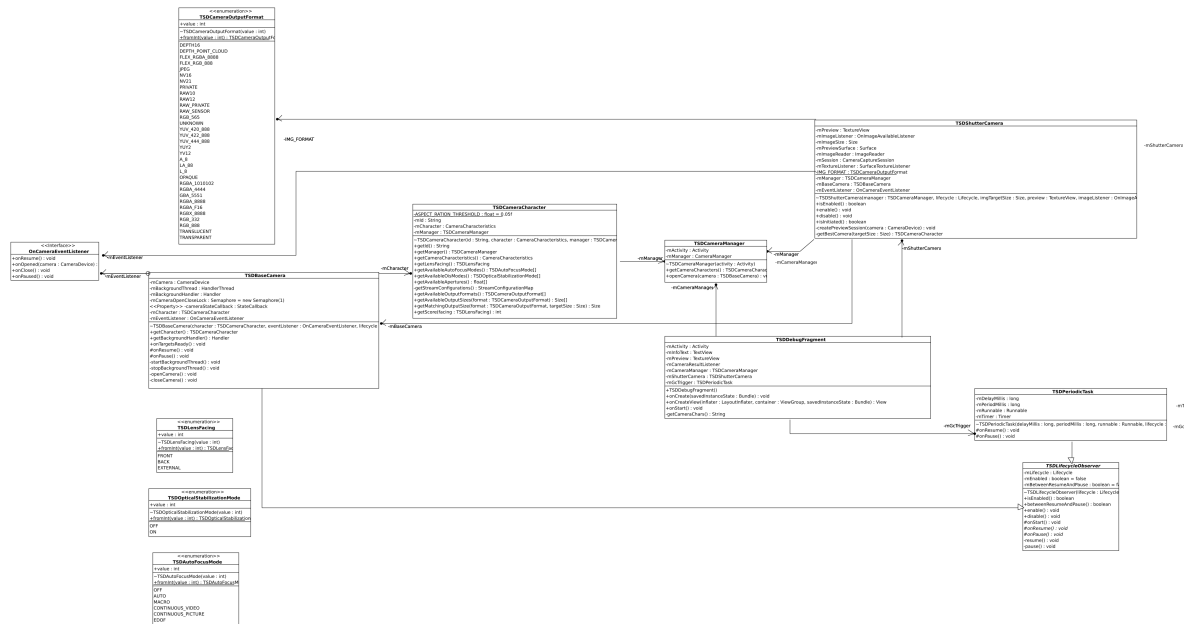
## 6.13 `de.swp.tsd.trafficsigndetection.TSDCameraManager` Class Reference

### Public Member Functions

- [TSDCameraCharacter \[\]](#) [getCameraCharacters](#) ()
- void [openCamera](#) ([TSDBaseCamera](#) camera)

- **TSDCameraManager** (Activity activity)

**TSDCameraManager** is a Wrapper around the **CameraManager**. It adds some implementations to make it compatible with the other wrappers **TSDCameraCharacter** and **TSDBaseCamera**.



#### 6.13.2.1 TSDCameraManager()

Creates a new instance of CameraDetector preparing for loading of camera information.

<i>activity</i>	The Activity running in
-----------------	-------------------------

Generated by Doxygen

### 6.13.3.1 `getCameraCharacters()`

```
TSDCameraCharacter [] de.swp.tsd.trafficsigndetection.TSDCameraManager.getCameraCharacters ( )
[inline]
```

Queries the characters of camera devices which are available to this device.

#### Returns

Array of all camera characters. If there are no cameras accessible or an error occurs this can be empty.

References `de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getCameraCharacteristics()`.

### 6.13.3.2 `openCamera()`

```
void de.swp.tsd.trafficsigndetection.TSDCameraManager.openCamera (
    TSDBaseCamera camera ) [inline]
```

This method tries to open the given camera instance. Please mind, that this does not include the permission request. This must be done before, otherwise the opening will fail. When the opening was successful, then `TSDBaseCamera#mCameraStateCallback` will be signaled which again will signal the `TSDBaseCamera.OnCameraEventListener` given to this camera instance.

#### Parameters

<i>camera</i>	Camera instance to open
---------------	-------------------------

References `de.swp.tsd.trafficsigndetection.TSDBaseCamera.getBackgroundHandler()`, `de.swp.tsd.trafficsigndetection.TSDBaseCamera.getCameraStateCallback()`, `de.swp.tsd.trafficsigndetection.TSDBaseCamera.getCharacter()`, and `de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getId()`.

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

- `App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraManager.java`

## 6.14 `de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat` Enum Reference

### Public Member Functions

- `TSDCameraOutputFormat` (int *value*)

### Static Public Member Functions

- static `TSDCameraOutputFormat fromInt` (int *value*)

## Public Attributes

- [DEPTH16](#) =(1144402265)
- [DEPTH\\_POINT\\_CLOUD](#) =(257)
- [FLEX\\_RGBA\\_8888](#) =(42)
- [FLEX\\_RGB\\_888](#) =(41)
- [JPEG](#) =(256)
- [NV16](#) =(16)
- [NV21](#) =(17)
- [PRIVATE](#) =(34)
- [RAW10](#) =(37)
- [RAW12](#) =(38)
- [RAW\\_PRIVATE](#) =(36)
- [RAW\\_SENSOR](#) =(32)
- [RGB\\_565](#) =(4)
- [UNKNOWN](#) =(0)
- [YUV\\_420\\_888](#) =(35)
- [YUV\\_422\\_888](#) =(39)
- [YUV\\_444\\_888](#) =(40)
- [YUY2](#) =(20)
- [YV12](#) =(842094169)
- [A\\_8](#) =(8)
- [LA\\_88](#) =(10)
- [L\\_8](#) =(9)
- [OPAQUE](#) =(-1)
- [RGBA\\_1010102](#) =(43)
- [RGBA\\_4444](#) =(7)
- [GBA\\_5551](#) =(6)
- [RGBA\\_8888](#) =(1)
- [RGBA\\_F16](#) =(22)
- [RGBX\\_8888](#) =(2)
- [RGB\\_332](#) =(11)
- [RGB\\_888](#) =(3)
- [TRANSLUCENT](#) =(-3)
- [TRANSPARENT](#) =(-2)
- final int [value](#)

### 6.14.1 Detailed Description

Describes the possible output formats from the camera device according to the Android API Documents for [ImageFormat](#) and [PixelFormat](#). For detailed descriptions of the enum values please visit these documents.

### 6.14.2 Constructor & Destructor Documentation

#### 6.14.2.1 TSDCameraOutputFormat()

```
de.swp.tsd.trafficsignetection.TSDCameraOutputFormat.TSDCameraOutputFormat (
    int value ) [inline]
```

This constructor creates a new enum instance setting the internal enum value to the given one.

#### Parameters

<i>value</i>	This value should match the value of the constant in the Android API describing the same format.
--------------	--

### 6.14.3 Member Function Documentation

#### 6.14.3.1 fromInt()

```
static TSDCameraOutputFormat de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.fromInt (
    int value ) [inline], [static]
```

This function can convert a given value which is one of the constant describing image or pixel formats in the Android API to the corresponding enum value. If the given value does not match any defined enum, then UNKNOWN will be returned.

#### Parameters

<i>value</i>	Integer that should be converted.
--------------	-----------------------------------

#### Returns

Enum value that corresponds to the given int value. If not matching one is found, UNKNOWN will be returned.

### 6.14.4 Member Data Documentation

#### 6.14.4.1 A\_8

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.A_8 =(8)
```

#### 6.14.4.2 DEPTH16

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.DEPTH16 =(1144402265)
```

#### 6.14.4.3 DEPTH\_POINT\_CLOUD

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.DEPTH_POINT_CLOUD =(257)
```

#### 6.14.4.4 FLEX\_RGB\_888

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.FLEX_RGB_888 =(41)
```

#### 6.14.4.5 FLEX\_RGBA\_8888

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.FLEX_RGBA_8888 =(42)
```

#### 6.14.4.6 GBA\_5551

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.GBA_5551 =(6)
```

#### 6.14.4.7 JPEG

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.JPEG =(256)
```

#### 6.14.4.8 L\_8

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.L_8 =(9)
```

#### 6.14.4.9 LA\_88

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.LA_88 =(10)
```

#### 6.14.4.10 NV16

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.NV16 =(16)
```

#### 6.14.4.11 NV21

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.NV21 =(17)
```

#### 6.14.4.12 OPAQUE

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.OPAQUE = (-1)
```

#### 6.14.4.13 PRIVATE

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.PRIVATE = (34)
```

#### 6.14.4.14 RAW10

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RAW10 = (37)
```

#### 6.14.4.15 RAW12

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RAW12 = (38)
```

#### 6.14.4.16 RAW\_PRIVATE

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RAW_PRIVATE = (36)
```

#### 6.14.4.17 RAW\_SENSOR

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RAW_SENSOR = (32)
```

#### 6.14.4.18 RGB\_332

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RGB_332 = (11)
```

#### 6.14.4.19 RGB\_565

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RGB_565 = (4)
```



**6.14.4.20 RGB\_888**

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RGB_888 = (3)
```

**6.14.4.21 RGBA\_1010102**

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RGBA_1010102 = (43)
```

**6.14.4.22 RGBA\_4444**

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RGBA_4444 = (7)
```

**6.14.4.23 RGBA\_8888**

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RGBA_8888 = (1)
```

**6.14.4.24 RGBA\_F16**

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RGBA_F16 = (22)
```

**6.14.4.25 RGBX\_8888**

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.RGBX_8888 = (2)
```

**6.14.4.26 TRANSLUCENT**

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.TRANSLUCENT = (-3)
```

**6.14.4.27 TRANSPARENT**

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.TRANSPARENT = (-2)
```

#### 6.14.4.28 UNKNOWN

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.UNKNOWN = (0)
```

#### 6.14.4.29 value

```
final int de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.value
```

This value describes the value of the constant in the Android API describing the same format this enum wraps.

#### 6.14.4.30 YUV\_420\_888

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.YUV_420_888 = (35)
```

#### 6.14.4.31 YUV\_422\_888

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.YUV_422_888 = (39)
```

#### 6.14.4.32 YUV\_444\_888

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.YUV_444_888 = (40)
```

#### 6.14.4.33 YUY2

```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.YUY2 = (20)
```

#### 6.14.4.34 YV12

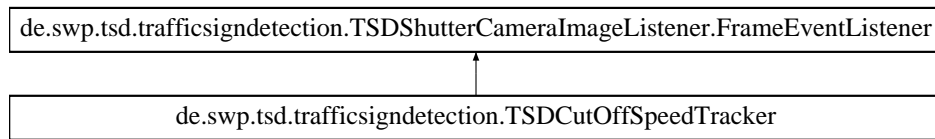
```
de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.YV12 = (842094169)
```

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraOutputFormat.java](#)

## 6.15 de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker:



### Classes

- interface [OnCutOffSpeedCalculatedListener](#)

### Public Member Functions

- void [setCameraViewAngle](#) (float cameraViewAngle)
- void [onFrameFinished](#) (long duration)

### Package Functions

- [TSDCutOffSpeedTracker](#) (@NonNull [OnCutOffSpeedCalculatedListener](#) listener, float cameraViewAngle)

### 6.15.1 Detailed Description

This class defines an [TSDShutterCameraImageListener.FrameEventListener](#) which uses the given information to calculate an approximation of the maximum speed (cut-off speed) which should still allow the device to properly detect the signs on the captured images. The background to this is, that sign detection is extremely processing intensive and pushes even high-end phones from two years ago to the limits. That is why this class tries to use the given time which was taken to process single frames and further information to calculate an approximated value describing the maximum speed the device can move and still should be fast enough to capture and process enough frames to get every sign around there.

### 6.15.2 Constructor & Destructor Documentation

#### 6.15.2.1 TSDCutOffSpeedTracker()

```

de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.TSDCutOffSpeedTracker (
    @NonNull OnCutOffSpeedCalculatedListener listener,
    float cameraViewAngle ) [inline], [package]
  
```

This constructor creates a new instance of [TSDCutOffSpeedTracker](#).

## Parameters

<i>listener</i>	Observer which gets notified when a new cut-off speed was calculated.
<i>cameraViewAngle</i>	View angle of the camera in use to capture the frames in radians.

References [de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.setCameraViewAngle\(\)](#).

### 6.15.3 Member Function Documentation

#### 6.15.3.1 onFrameFinished()

```
void de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.onFrameFinished (
    long duration ) [inline]
```

This method is called every time one camera frame is finished. The given processing time and image width is used to calculate an approximated value for the cut-off speed described in [TSDCutOffSpeedTracker](#). The calculated value is buffered inside an ring memory and an average of the last values is signaled to the given listener.

The calculation is as follows:

The maximum currently acceptable speed would be this, that would let each sign at least one time to be detected on a captured image. Taking the currently known processing time for one frame and the maximum distance a sign can be away while still getting detected we can calculate this speed by the simple speed formula  $v = s / t$ . But to give a little threshold to not depend on edge cases we multiply the distance by a threshold value `TSDCutOffSpeedTracker#DISTANCE_THRESHOLD_FACTOR` a little smaller than 1.

## Parameters

<i>duration</i>	Time it took to process the frame in milliseconds.
-----------------	--

Implements [de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.FrameEventListener](#).

References [de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.add\(\)](#), and [de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.getMaxSize\(\)](#).

#### 6.15.3.2 setCameraViewAngle()

```
void de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.setCameraViewAngle (
    float cameraViewAngle ) [inline]
```

Setter for the camera view angle. This will calculate and save the corresponding `TSDCutOffSpeedTracker#mLensFactor` and `TSDCutOffSpeedTracker#mMaxDistance`.

## Parameters

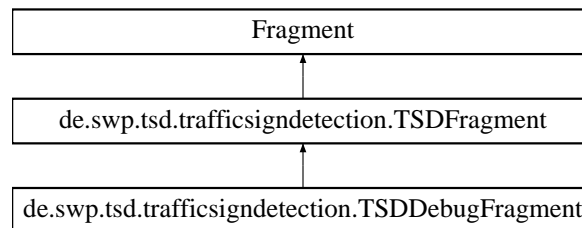
<i>cameraViewAngle</i>	Maximum view angle the camera supports in radians.
------------------------	--

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCutOffSpeedTracker.java](#)

## 6.16 de.swp.tsd.trafficsigndetection.TSDDebugFragment Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDDebugFragment:



### Public Member Functions

- [TSDDebugFragment \(\)](#)
- void [onCreate](#) (Bundle savedInstanceState)
- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void [onStart](#) ()

### Additional Inherited Members

#### 6.16.1 Detailed Description

This [TSDFragment](#) is used for debugging purposes and will get removed in the final product. It currently prints some information to every camera in the device to screen. Furthermore does it open a [TSDShutterCamera](#) with currently only activated live post view which displays the results of the TrafficSignAPI to screen, which processes live frames taken by the camera. This requires the TrafficSignAPI to already having been initialized. The fragment triggers the required permission requests and includes an [TSDPeriodicTask](#) which periodically triggers the GC to clean up rubbish produced by the camera beginning with Android Version 8.0 .

#### 6.16.2 Constructor & Destructor Documentation

### 6.16.2.1 TSDDebugFragment()

```
de.swp.tsd.trafficsigndetection.TSDDebugFragment.TSDDebugFragment ( ) [inline]
```

This constructor creates a new instance of [TSDDebugFragment](#) setting the correct values for [display name](#) and [display icon](#).

## 6.16.3 Member Function Documentation

### 6.16.3.1 onCreate()

```
void de.swp.tsd.trafficsigndetection.TSDDebugFragment.onCreate (
    Bundle savedInstanceState ) [inline]
```

This method gets called when the fragment is created. It initializes the periodic task to trigger the GC for cleaning up camera data rubbish.

#### Parameters

<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.
---------------------------	--

References [de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.disable\(\)](#).

### 6.16.3.2 onCreateView()

```
View de.swp.tsd.trafficsigndetection.TSDDebugFragment.onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState ) [inline]
```

This method gets called every time the view of the [TSDFragment](#) is newly created. It inflates the `fragment_debug` layout as user interface for the [TSDFragment](#). It initializes the shutter camera and sets the required views. The Shutter camera currently only displays to the `postView` of this ui layout. Preview and sign history are disabled.

#### Parameters

<i>inflater</i>	Inflater to use for inflating the user interface layout of this fragment.
<i>container</i>	The place the ui should be inflated to, but not attached.
<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.

**Returns**

The inflated view containing the user interface of this [TSDFragment](#).

References [de.swp.tsd.trafficsigndetection.TSDShutterCamera.disable\(\)](#), [de.swp.tsd.trafficsigndetection.TSDShutterCamera.getSensorRotation\(\)](#), and [de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.setSensorRotation\(\)](#).

**6.16.3.3 onStart()**

```
void de.swp.tsd.trafficsigndetection.TSDDebugFragment.onStart ( ) [inline]
```

This method gets called every time the [TSDFragment](#) is started. It asks for the required permissions and if those are granted enables the already in [TSDDebugFragment#onCreateView\(LayoutInflater, ViewGroup, Bundle\)](#) created [TSDShutterCamera](#).

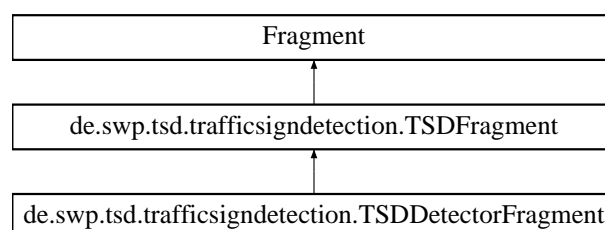
References [de.swp.tsd.trafficsigndetection.TSDLensFacing.BACK](#), [de.swp.tsd.trafficsigndetection.TSDCameraManager.getCameraCharacters\(\)](#), [de.swp.tsd.trafficsigndetection.TSDPermissionManager.getPermissions\(\)](#), and [de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat.JPEG](#).

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDebugFragment.java](#)

**6.17 de.swp.tsd.trafficsigndetection.TSDDetectorFragment Class Reference**

Inheritance diagram for [de.swp.tsd.trafficsigndetection.TSDDetectorFragment](#):

**Public Member Functions**

- [TSDDetectorFragment \(\)](#)
- void [onCreate](#) (Bundle savedInstanceState)
- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void [onStart](#) ()





## Parameters

<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.
---------------------------	--

References [de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.disable\(\)](#).

## 6.17.3.2 onCreateView()

```
View de.swp.tsd.trafficsigndetection.TSDDetectorFragment.onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState ) [inline]
```

This method gets called every time the view of the [TSDFragment](#) is newly created. It inflates the `fragment_detector` layout as user interface for the [TSDFragment](#). It creates and initializes the [TSDSignHistory](#), the `RecyclerView` and the main sign view to `isRelevant` this history. Furthermore the [TSDSpeedTracker](#) and the [TSDShutterCamera](#) get initialized. There is also a [TSDPeriodicTask](#) to refresh the time `isRelevant` for all visible [TSDSignCombinations](#) on screen.

## See also

[TSDDetectorFragment](#)

## Parameters

<i>inflater</i>	Inflater to use for inflating the user interface layout of this fragment.
<i>container</i>	The place the ui should be inflated to, but not attached.
<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.

## Returns

The inflated view containing the user interface of this [TSDFragment](#).

References [de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.disable\(\)](#), [de.swp.tsd.trafficsigndetection.TSDShutterCamera.disable\(\)](#), [de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displayAll\(\)](#), [de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displaySpeed\(\)](#), [de.swp.tsd.trafficsigndetection.TSDShutterCamera.getCameraCharacter\(\)](#), [de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getMaximumViewAngle\(\)](#), [de.swp.tsd.trafficsigndetection.TSDShutterCamera.getSensorRotation\(\)](#), [de.swp.tsd.trafficsigndetection.TSDSignHistory.getSignCombinations\(\)](#), [de.swp.tsd.trafficsigndetection.TSDSignHistory.isSpeedOk\(\)](#), [de.swp.tsd.trafficsigndetection.TSDHelper.readBooleanFromSharedPreferences\(\)](#), [de.swp.tsd.trafficsigndetection.TSDHelper.readIntegerFromSharedPreferences\(\)](#), [de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.setCameraViewAngle\(\)](#), [de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.setMaxSize\(\)](#), [de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.setSensorRotation\(\)](#), and [de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.setWarningTransparency\(\)](#).

### 6.17.3.3 onStart()

```
void de.swp.tsd.trafficsigndetection.TSDDetectorFragment.onStart ( ) [inline]
```

This method gets called every time the [TSDFragment](#) is started. It asks for the required camera permissions and enables the TSDDetectorFragment#mShutterCamera if those were granted. If the camera was granted, then also the location permission gets asked for and if this one is also granted, the TSDDetectorFragment#mSpeedTracker gets enabled, too. If one of the permissions does not get granted, the connected objects will not get started. The user will get an information about the consequences of his decision and the possibility to get another ask for the permissions.

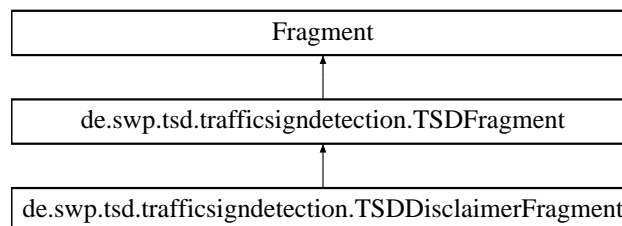
References `de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.enable()`, and `de.swp.tsd.trafficsigndetection.TSDPermissionManager.getPermissions()`.

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

- `App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDetectorFragment.java`

## 6.18 de.swp.tsd.trafficsigndetection.TSDDisclaimerFragment Class Reference

Inheritance diagram for `de.swp.tsd.trafficsigndetection.TSDDisclaimerFragment`:



### Public Member Functions

- [TSDDisclaimerFragment](#) ()
- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

### Additional Inherited Members

#### 6.18.1 Detailed Description

This [TSDFragment](#) displays the user the disclaimer of the app. The user will not be able to continue with the application without accepting the disclaimer. It will be shown on every startup of the app.

#### 6.18.2 Constructor & Destructor Documentation

### 6.18.2.1 TSDDisclaimerFragment()

```
de.swp.tsd.trafficsigndetection.TSDDisclaimerFragment.TSDDisclaimerFragment ( ) [inline]
```

This constructor creates a new instance of [TSDDisclaimerFragment](#) setting the correct values for [isRelevant name](#) and [isRelevant icon](#).

## 6.18.3 Member Function Documentation

### 6.18.3.1 onCreateView()

```
View de.swp.tsd.trafficsigndetection.TSDDisclaimerFragment.onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState ) [inline]
```

This method gets called every time the view of the [TSDFragment](#) is newly created. It inflates the fragment\_disclaimer layout as user interface for the [TSDFragment](#).

See also

[TSDDisclaimerFragment](#)

#### Parameters

<i>inflater</i>	Inflater to use for inflating the user interface layout of this fragment.
<i>container</i>	The place the ui should be inflated to, but not attached.
<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.

#### Returns

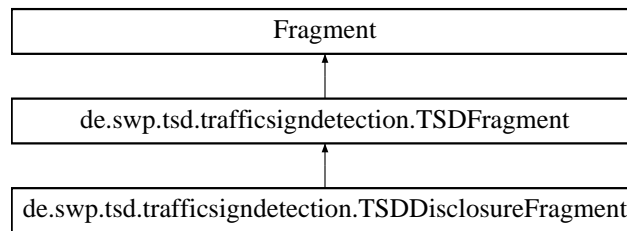
The inflated view containing the user interface of this [TSDFragment](#).

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDDisclaimerFragment.java](#)

## 6.19 de.swp.tsd.trafficsigndetection.TSDDisclosureFragment Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDDisclosureFragment:



## Public Member Functions

- [TSDDisclosureFragment\(\)](#)
- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

## Additional Inherited Members

### 6.19.1 Detailed Description

This [TSDFragment](#) displays the user the disclosure of the app. The user will not be able to continue with the application without accepting the disclosure. It will be shown at least once at the first start and then every start when the user is likely to come across a permission request.

### 6.19.2 Constructor & Destructor Documentation

#### 6.19.2.1 TSDDisclosureFragment()

```
de.swp.tsd.trafficsigndetection.TSDDisclosureFragment.TSDDisclosureFragment ( ) [inline]
```

This constructor creates a new instance of [TSDDisclosureFragment](#) setting the correct values for [isRelevant name](#) and [isRelevant icon](#).

### 6.19.3 Member Function Documentation

#### 6.19.3.1 onCreateView()

```
View de.swp.tsd.trafficsigndetection.TSDDisclosureFragment.onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState ) [inline]
```

This method gets called every time the view of the [TSDFragment](#) is newly created. It inflates the `fragment_disclosure` layout as user interface for the [TSDFragment](#).

See also

[TSDDisclosureFragment](#)

## Parameters

<i>inflater</i>	Inflater to use for inflating the user interface layout of this fragment.
<i>container</i>	The place the ui should be inflated to, but not attached.
<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.

## Returns

The inflated view containing the user interface of this [TSDFragment](#).

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsign detection/TSDDisclosureFragment.java](#)

## 6.20 de.swp.tsd.trafficsign detection.TSDDrawable Class Reference

### Public Member Functions

- Integer [getDrawableId](#) ()
- Drawable [getDrawable](#) ()
- int [getWidth](#) ()
- int [getHeight](#) ()
- boolean [equals](#) (Object obj)

### Package Functions

- [TSDDrawable](#) (Integer drawableId, @NonNull Drawable drawable)
- [TSDDrawable](#) (Integer drawableId, Drawable drawable, int width, int height)

### 6.20.1 Detailed Description

This class wraps a drawable loaded from resources or from somewhere else and additional information like the resource id, the image width and the image height. Since the actual width and height of the image can be overridden in one constructor, those information should only be used for scaling, not for direct access. To get the actual dimensions, they need to be directly queried from the drawable instance.

### 6.20.2 Constructor & Destructor Documentation

#### 6.20.2.1 TSDDrawable() [1/2]

```
de.swp.tsd.trafficsign detection.TSDDrawable.TSDDrawable (
    Integer drawableId,
    @NonNull Drawable drawable ) [inline], [package]
```

This constructor creates a new instance of [TSDDrawable](#) querying for the actual width and height of the image.

## Parameters

<i>drawable↔ Id</i>	Actual resource id of the drawable. If the drawable was not loaded from resources, then this should be null.
<i>drawable</i>	Drawable instance also the width and height will be taken from.

## 6.20.2.2 TSDDrawable() [2/2]

```
de.swp.tsd.trafficsigndetection.TSDDrawable.TSDDrawable (
    Integer drawableId,
    Drawable drawable,
    int width,
    int height ) [inline], [package]
```

This constructor creates a new instance of [TSDDrawable](#) overriding the actual width and height of the image. This can be used when the image should be scaled like if it had different dimensions.

## Parameters

<i>drawable↔ Id</i>	Actual resource id of the drawable. If the drawable was not loaded from resources, then this should be null.
<i>drawable</i>	Drawable loaded from resources.
<i>width</i>	Width to save as this image width.
<i>height</i>	Height to save as this image height.

## 6.20.3 Member Function Documentation

## 6.20.3.1 equals()

```
boolean de.swp.tsd.trafficsigndetection.TSDDrawable.equals (
    Object obj ) [inline]
```

This method can be used to compare two [TSDDrawable](#) instances. Two [TSDDrawable](#) instances match, if the following information are equal: the id, the given width and given height and the actual width and height of the drawable.

## Parameters

<i>obj</i>	Other <a href="#">TSDDrawable</a> to compare to.
------------	--

## Returns

The result of the comparison or false, if the given object to compare to was not of the same class as this instance.

References `de.swp.tsd.trafficsigndetection.TSDHelper.equalNullable()`, and `de.swp.tsd.trafficsigndetection.TSDDrawable.TSDDrawable()`.

### 6.20.3.2 `getDrawable()`

```
Drawable de.swp.tsd.trafficsigndetection.TSDDrawable.getDrawable ( ) [inline]
```

Getter for the drawable.

#### Returns

Drawable instance

### 6.20.3.3 `getDrawableId()`

```
Integer de.swp.tsd.trafficsigndetection.TSDDrawable.getDrawableId ( ) [inline]
```

Getter for the drawable resource id. If the drawable was not loaded from resources, then this will be null.

#### Returns

Drawable resource id or null, if the drawable was not loaded from resources.

### 6.20.3.4 `getHeight()`

```
int de.swp.tsd.trafficsigndetection.TSDDrawable.getHeight ( ) [inline]
```

Getter for the height of the drawable. Please mind, that this value could be overridden in the constructor and then not match the actual height of the image. Only use this for scaling.

#### Returns

Height of the drawable

### 6.20.3.5 `getWidth()`

```
int de.swp.tsd.trafficsigndetection.TSDDrawable.getWidth ( ) [inline]
```

Getter for the width of the drawable. Please mind, that this value could be overridden in the constructor and then not match the actual width of the image. Only use this for scaling.

#### Returns

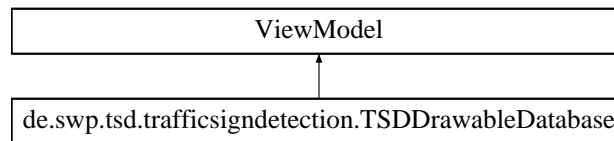
Width of the drawable

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

- `App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDrawable.java`

## 6.21 de.swp.tsd.trafficsigndetection.TSDDrawableDatabase Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDDrawableDatabase:



### Public Member Functions

- [TSDDrawable getDrawable](#) (Integer drawableId, Context context)

#### 6.21.1 Detailed Description

This class acts as a collection for loaded drawables. When loading a drawable via an instance of this class, then if already loaded at some point it will be returned back without further loading, otherwise it will be loaded and stored for later access. The identification is done by the given resource id of the drawable. If the drawable is not already in here, then it will get loaded from the resources using the given id.

#### 6.21.2 Member Function Documentation

##### 6.21.2.1 getDrawable()

```

TSDDrawable de.swp.tsd.trafficsigndetection.TSDDrawableDatabase.getDrawable (
    Integer drawableId,
    Context context ) [inline]
  
```

This method queries for a drawable by its resource id. If the drawable was already loaded, then this instance will get returned, otherwise it will be loaded from the resources and stored for later usage. If the id is null, then null will be returned. This can be used in combination with application side created drawables and loading logics which work after having called this method.

##### Parameters

<i>drawableId</i>	Resource id of the drawable or null, if the wanted drawable is not in resources.
<i>context</i>	Context to load from the resources.

##### Returns

Drawable if it could be loaded or if it was already in here, otherwise null.

References `de.swp.tsd.trafficsigndetection.TSDDrawable.getDrawable()`.

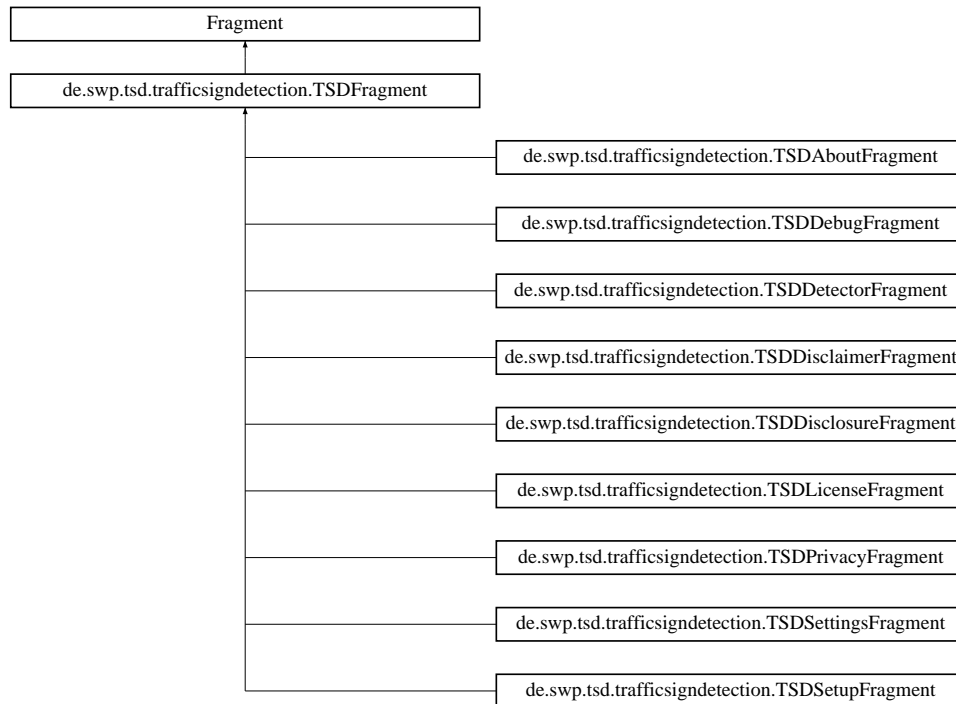
The documentation for this class was generated from the following file:



- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDrawableDatabase.java](#)

## 6.22 de.swp.tsd.trafficsigndetection.TSDFragment Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDFragment:



### Public Member Functions

- [TSDFragment](#) (int [displayNameId](#), int [displayIconId](#))

### Public Attributes

- final int [displayNameId](#)
- final int [displayIconId](#)

#### 6.22.1 Detailed Description

This abstract class defines the base for every Fragment that will should be accessible through the navigation drawer. It includes additional information about the name and the icon the navigation drawer should isRelevant for this fragment.

#### 6.22.2 Constructor & Destructor Documentation

#### 6.22.2.1 TSDFragment()

```
de.swp.tsd.trafficsigndetection.TSDFragment.TSDFragment (
    int displayNameId,
    int displayIconId ) [inline]
```

Creates a new instance of an [TSDFragment](#) defining isRelevant information for the navigation drawer.

## Parameters

<i>display↔ NameId</i>	Defines the resource id for the string that will be displayed in the navigation drawer.
<i>displayIconId</i>	Defines the Drawable that will be displayed as icon in the navigation drawer.

References `de.swp.tsd.trafficsigndetection.TSDFragment.displayIconId`, and `de.swp.tsd.trafficsigndetection.TSD↔  
Fragment.displayNameId`.

### 6.22.3 Member Data Documentation

#### 6.22.3.1 displayIconId

```
final int de.swp.tsd.trafficsigndetection.TSDFragment.displayIconId
```

Defines the resource id for the Drawable that will be displayed as icon in the navigation drawer.

#### 6.22.3.2 displayNameId

```
final int de.swp.tsd.trafficsigndetection.TSDFragment.displayNameId
```

Defines the resource id for the string that will be displayed in the navigation drawer.

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

- `App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDFragment.java`

## 6.23 de.swp.tsd.trafficsigndetection.TSDHelper Class Reference

### Static Public Member Functions

- static int `dpToPx` (int dp, Context context)
- static< T > boolean `equalNullable` (T a, T b)
- static void `writeBooleanToSharedPreferences` (int keyStringId, boolean value, @NonNull Context context)
- static boolean `readBooleanFromSharedPreferences` (int keyStringId, boolean defaultValue, @NonNull Context context)
- static void `writeIntegerToSharedPreferences` (int keyStringId, int value, @NonNull Context context)
- static int `readIntegerFromSharedPreferences` (int keyStringId, int defaultValue, @NonNull Context context)

#### 6.23.1 Detailed Description

This class holds some helper functions and cannot be instantiated.

## 6.23.2 Member Function Documentation

### 6.23.2.1 dpToPx()

```
static int de.swp.tsd.trafficsigndetection.TSDHelper.dpToPx (  
    int dp,  
    Context context )    [inline], [static]
```

This function converts a given dimension measure from dp into px using the display metrics of the device.

## Parameters

<i>dp</i>	Dimension measure in dp.
<i>context</i>	Context used to get display metrics.

## Returns

Dimension measure in px.

## 6.23.2.2 equalNullable()

```
static <T> boolean de.swp.tsd.trafficsigndetection.TSDHelper.equalNullable (  
    T a,  
    T b ) [inline], [static]
```

This function compares two values for equality which can be null. If both are null, they are also equal.

## Parameters

<i>a</i>	First value
<i>b</i>	Second value
<i>&lt;T&gt;</i>	Type of teh values

## Returns

Whether the values are equal or not.

## 6.23.2.3 readBooleanFromSharedPreferences()

```
static boolean de.swp.tsd.trafficsigndetection.TSDHelper.readBooleanFromSharedPreferences (  
    int keyStringId,  
    boolean defaultValue,  
    @NonNull Context context ) [inline], [static]
```

This function reads the boolean value from the shared preferences at the given key.

## Parameters

<i>keyStringId</i>	Resource id of the key string.
<i>defaultValue</i>	Value to return if the key was not found.
<i>context</i>	Context for shared preferences.

**Returns**

The boolean value if it was found or the default value.

**6.23.2.4 readIntegerFromSharedPreferences()**

```
static int de.swp.tsd.trafficsigndetection.TSDHelper.readIntegerFromSharedPreferences (
    int keyStringId,
    int defaultValue,
    @NonNull Context context ) [inline], [static]
```

This function reads the integer value from the shared preferences at the given key.

**Parameters**

<i>keyStringId</i>	Resource id of the key string.
<i>defaultValue</i>	Value to return if the key was not found.
<i>context</i>	Context for shared preferences.

**Returns**

The boolean value if it was found or the default value.

**6.23.2.5 writeBooleanToSharedPreferences()**

```
static void de.swp.tsd.trafficsigndetection.TSDHelper.writeBooleanToSharedPreferences (
    int keyStringId,
    boolean value,
    @NonNull Context context ) [inline], [static]
```

This function writes the given boolean value at the given key to the shared preferences. This way the boolean is saved for access even after restarts of the app.

**Parameters**

<i>key↔StringId</i>	Resource id of the key string.
<i>value</i>	Value to write.
<i>context</i>	Context for shared preferences.

**6.23.2.6 writeIntegerToSharedPreferences()**

```
static void de.swp.tsd.trafficsigndetection.TSDHelper.writeIntegerToSharedPreferences (
    int keyStringId,
```

```
int value,
@NonNull Context context ) [inline], [static]
```

This function writes the given integer value at the given key to the shared preferences. This way the integer is saved for access even after restarts of the app.

#### Parameters

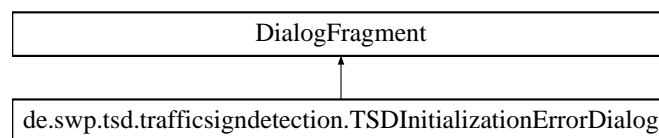
<i>key</i> ↔ <i>StringId</i>	Resource id of the key string.
<i>value</i>	Value to write.
<i>context</i>	Context for shared preferences.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDHelper.java](#)

## 6.24 de.swp.tsd.trafficsigndetection.TSDInitializationErrorDialog Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDInitializationErrorDialog:



### Public Member Functions

- Dialog [onCreateDialog](#) (Bundle savedInstanceState)

#### 6.24.1 Detailed Description

This class defines a fragment dialog showing an error happened during initialization of the api. This dialog is shown at startup of the application if api initialization failed and is used therefore in the [MainActivity](#). The error dialog is just a box with the in the resources saved error text and an button to accept it.

#### 6.24.2 Member Function Documentation

### 6.24.2.1 onCreateDialog()

```
Dialog de.swp.tsd.trafficsigndetection.TSDInitializationErrorDialog.onCreateDialog (
    Bundle savedInstanceState ) [inline]
```

This function gets called to create the dialog. It sets the error text defined in legal\_disclaimer in resources and the accept button.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDInitializationErrorDialog.java](#)

## 6.25 de.swp.tsd.trafficsigndetection.TSDLensFacing Enum Reference

### Public Member Functions

- [TSDLensFacing](#) (int [value](#))

### Static Public Member Functions

- static [TSDLensFacing fromInt](#) (int [value](#))

### Public Attributes

- [FRONT](#) =(0)
- [BACK](#) =(1)
- [EXTERNAL](#) =(2)
- final int [value](#)

### 6.25.1 Detailed Description

Describes wraps the possible lens facing direction of the camera instance according to the [Android API](#) into an enum. See the documentation for detailed information.

### 6.25.2 Constructor & Destructor Documentation

#### 6.25.2.1 TSDLensFacing()

```
de.swp.tsd.trafficsigndetection.TSDLensFacing.TSDLensFacing (
    int value ) [inline]
```

This constructor creates a new enum instance setting the internal enum value to the given one.



#### Parameters

<i>value</i>	This value should match the value of the constant in the Android API describing the same direction.
--------------	---

### 6.25.3 Member Function Documentation

#### 6.25.3.1 fromInt()

```
static TSDLensFacing de.swp.tsd.trafficsigndetection.TSDLensFacing.fromInt (
    int value ) [inline], [static]
```

This function can convert a given value which is one of the constant describing lens facing directions in the Android API to the corresponding enum value. If the given value does not match any defined enum, then EXTERNAL will be returned.

#### Parameters

<i>value</i>	Integer that should be converted.
--------------	-----------------------------------

#### Returns

Enum value that corresponds to the given int value. If not matching one is found, EXTERNAL will be returned.

### 6.25.4 Member Data Documentation

#### 6.25.4.1 BACK

```
de.swp.tsd.trafficsigndetection.TSDLensFacing.BACK =(1)
```

The camera is located at the back of the device. [LENS\\_FACING\\_BACK](#)

#### 6.25.4.2 EXTERNAL

```
de.swp.tsd.trafficsigndetection.TSDLensFacing.EXTERNAL =(2)
```

The camera is located externally of the device. [LENS\\_FACING\\_EXTERNAL](#)

#### 6.25.4.3 FRONT

```
de.swp.tsd.trafficsigndetection.TSDLensFacing.FRONT =(0)
```

The camera is located at the front of the device. [LENS\\_FACING\\_FRONT](#)

#### 6.25.4.4 value

```
final int de.swp.tsd.trafficsigndetection.TSDLensFacing.value
```

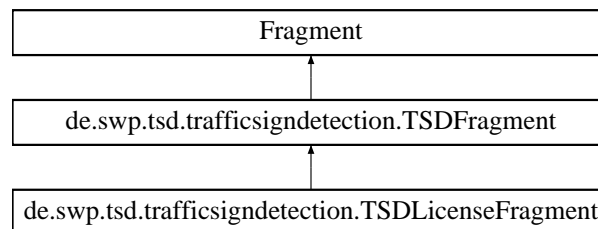
This value describes the value of the constant in the Android API describing the same direction this enum wraps.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDLensFacing.java](#)

## 6.26 de.swp.tsd.trafficsigndetection.TSDLicenseFragment Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDLicenseFragment:



### Public Member Functions

- [TSDLicenseFragment](#) ()
- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

### Additional Inherited Members

#### 6.26.1 Detailed Description

This [TSDFragment](#) displays the user the open source parts of this app and the corresponding licenses.

#### 6.26.2 Constructor & Destructor Documentation

##### 6.26.2.1 TSDLicenseFragment()

```
de.swp.tsd.trafficsigndetection.TSDLicenseFragment.TSDLicenseFragment ( ) [inline]
```

This constructor creates a new instance of [TSDLicenseFragment](#) setting the correct values for [display name](#) and [display icon](#).

## 6.26.3 Member Function Documentation

### 6.26.3.1 onCreateView()

```
View de.swp.tsd.trafficsigndetection.TSDLicenseFragment.onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState ) [inline]
```

This method gets called every time the view of the [TSDFragment](#) is newly created. It inflates the fragment\_license layout as user interface for the [TSDFragment](#).

See also

[TSDLicenseFragment](#)

#### Parameters

<i>inflater</i>	Inflater to use for inflating the user interface layout of this fragment.
<i>container</i>	The place the ui should be inflated to, but not attached.
<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.

#### Returns

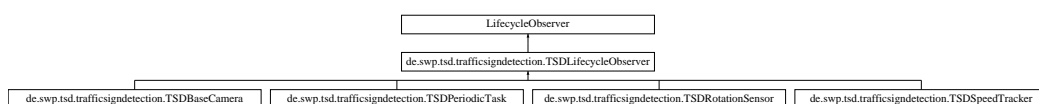
The inflated view containing the user interface of this [TSDFragment](#).

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDLicenseFragment.java](#)

## 6.27 de.swp.tsd.trafficsigndetection.TSDLifecycleObserver Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDLifecycleObserver:



### Public Member Functions

- boolean [isEnabled](#) ()
- boolean [betweenResumeAndPause](#) ()
- void [enable](#) ()
- void [disable](#) ()

## Protected Member Functions

- void [onStart](#) ()
- abstract void [onResume](#) ()
- abstract void [onPause](#) ()

## Package Functions

- [TSDLifecycleObserver](#) (@NonNull Lifecycle lifecycle)

### 6.27.1 Detailed Description

[TSDLifecycleObserver](#) is the base class for every lifecycle aware object in this project. It is implemented to pause and resume every child instance on ON\_PAUSE and ON\_RESUME events of the parent lifecycle. It also implements a possibility to disable the objects and to find out whether the object currently is between the [TSDLifecycleObserver#onResume\(\)](#) and [TSDLifecycleObserver#onPause\(\)](#) calls.

### 6.27.2 Constructor & Destructor Documentation

#### 6.27.2.1 TSDLifecycleObserver()

```
de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.TSDLifecycleObserver (
    @NonNull Lifecycle lifecycle ) [inline], [package]
```

This constructor creates a new instance of [TSDLifecycleObserver](#). This does not enable the object.

#### Parameters

<i>lifecycle</i>	Lifecycle of the parent
------------------	-------------------------

### 6.27.3 Member Function Documentation

#### 6.27.3.1 betweenResumeAndPause()

```
boolean de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.betweenResumeAndPause ( ) [inline]
```

This method queries whether the objects lifecycle currently is in between resume and pause events.

#### Returns

Boolean signaling whether the lifecycle is in between ON\_RESUME and ON\_PAUSE.

### 6.27.3.2 disable()

```
void de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.disable ( ) [inline]
```

This method clears the object if not already happened. It unregisters the object as observer for the given lifecycle and thus disables automatic start and stop on parent lifecycle PAUSE and RESUME.

References de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.onPause().

### 6.27.3.3 enable()

```
void de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.enable ( ) [inline]
```

This method starts the object if not already happened. It registers the object as observer for the given lifecycle and thus enables automatic start and stop on parent lifecycle PAUSE and RESUME. Then it also calls TSDLifecycleObserver#resume().

### 6.27.3.4 isEnabled()

```
boolean de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.isEnabled ( ) [inline]
```

This method queries whether the object should be running or not. When enabled, the object will get paused and resumed by the lifecycle of the parent.

#### Returns

Enable flag

### 6.27.3.5 onPause()

```
abstract void de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.onPause ( ) [abstract],  
[protected]
```

This method is called when the object should stop or pause working, when the parents lifecycle hits ON\_PAUSE and the object is enabled.

### 6.27.3.6 onResume()

```
abstract void de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.onResume ( ) [abstract],  
[protected]
```

This method is called when the object should continue working, when the parents lifecycle hits ON\_RESUME and the object is enabled or gets enabled.

### 6.27.3.7 onStart()

```
void de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.onStart ( ) [inline], [protected]
```

This method is called when the parent lifecycle hits the ON\_START event. This is called if enabled or not and could be used for permission requests.

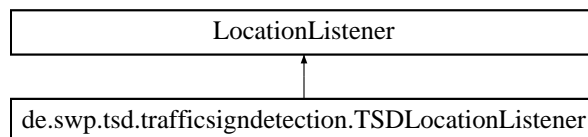
References `de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.onPause()`, and `de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.onResume()`.

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLifecycleObserver.java](#)

## 6.28 de.swp.tsd.trafficsigndetection.TSDLocationListener Class Reference

Inheritance diagram for `de.swp.tsd.trafficsigndetection.TSDLocationListener`:



### Public Member Functions

- void [onLocationChanged](#) (Location location)
- void [onStatusChanged](#) (String provider, int status, Bundle extras)
- void [onProviderEnabled](#) (String provider)
- void [onProviderDisabled](#) (String provider)

### Package Functions

- [TSDLocationListener](#) (TSDSpeedTracker.OnSpeedCalculatedListener listener)

### 6.28.1 Detailed Description

This class defines a `LocationListener` being signaled on every location event and signaling a [speed listener](#) with the internally calculated current speed.

### 6.28.2 Constructor & Destructor Documentation

#### 6.28.2.1 TSDLocationListener()

```
de.swp.tsd.trafficsigndetection.TSDLocationListener.TSDLocationListener (
    TSDSpeedTracker.OnSpeedCalculatedListener listener ) [inline], [package]
```

This constructor creates a new instance of [TSDLocationListener](#).

## Parameters

<i>listener</i>	Listener to signal the calculated current speed to.
-----------------	---

## 6.28.3 Member Function Documentation

## 6.28.3.1 onLocationChanged()

```
void de.swp.tsd.trafficsigndetection.TSDLocationListener.onLocationChanged (
    Location location ) [inline]
```

This method gets called every time the sensor registers a change in location. Each new location is saved inside the location ring memory, the average speed is calculated by TSDLocationListener#getAverageSpeed() and signaled to the speed listener. This could also be -1, if there are not enough location points collected in the memory.

## Parameters

<i>location</i>	New location determined by the location sensor.
-----------------	---

References de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.add().

## 6.28.3.2 onProviderDisabled()

```
void de.swp.tsd.trafficsigndetection.TSDLocationListener.onProviderDisabled (
    String provider ) [inline]
```

This method gets called when the provider delivering the location information gets disabled. This will clear the TSDLocationListener#mLocationList and signal -1 to the speed listener as current speed.

## Parameters

<i>provider</i>	Location provider getting disabled.
-----------------	-------------------------------------

References de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.getMaxSize().

## 6.28.3.3 onProviderEnabled()

```
void de.swp.tsd.trafficsigndetection.TSDLocationListener.onProviderEnabled (
    String provider ) [inline]
```

This method gets called when the provider delivering the location information gets enabled. Here this does nothing.

## Parameters

<i>provider</i>	Location provider getting enabled.
-----------------	------------------------------------

## 6.28.3.4 onStatusChanged()

```
void de.swp.tsd.trafficsigndetection.TSDLocationListener.onStatusChanged (
    String provider,
    int status,
    Bundle extras ) [inline]
```

This function gets called when the status of the location detection has changed. This will clear the TSDLocationListener#mLocationList and signal -1 to the speed listener as current speed.

## Parameters

<i>provider</i>	Location provider causing this status change.
<i>status</i>	The new status of location detection.
<i>extras</i>	Some extra information like number of satellites.

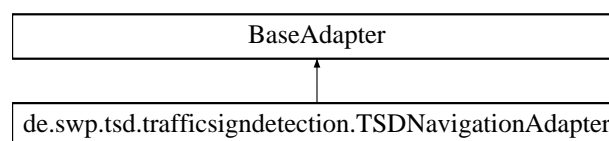
References de.swp.tsd.trafficsigndetection.TSDLocationListener.onProviderDisabled().

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDLocationListener.java](#)

## 6.29 de.swp.tsd.trafficsigndetection.TSDNavigationAdapter Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDNavigationAdapter:



## Public Member Functions

- int [getCount](#) ()
- Object [getItem](#) (int position)
- long [getItemId](#) (int position)
- View [getView](#) (int position, View convertView, ViewGroup parent)

## Package Functions

- [TSDNavigationAdapter](#) (Context context, [TSDFragment\[\]](#) fragments)



### 6.29.1 Detailed Description

This class is an adapter for displaying the individual items in the navigation menu. It takes [TSDFragments](#) as items and displays their titles and icons defined in there in a list inside the navigation menu.

### 6.29.2 Constructor & Destructor Documentation

#### 6.29.2.1 TSDNavigationAdapter()

```
de.swp.tsd.trafficsigndetection.TSDNavigationAdapter.TSDNavigationAdapter (
    Context context,
    TSDFragment [] fragments ) [inline], [package]
```

This constructor creates a new instance of [TSDNavigationAdapter](#) taking all the [TSDFragments](#) to display.

##### Parameters

<i>context</i>	Context the adapter is used in.
<i>fragments</i>	Array of the <a href="#">TSDFragments</a> that should be displayed as menu items. The order of the items is from array start to end displayed from top to bottom.

### 6.29.3 Member Function Documentation

#### 6.29.3.1 getCount()

```
int de.swp.tsd.trafficsigndetection.TSDNavigationAdapter.getCount ( ) [inline]
```

This method queries the number of menu items or [TSDFragments](#) to display.

##### Returns

The number of menu items or [TSDFragments](#) to display.

#### 6.29.3.2 getItem()

```
Object de.swp.tsd.trafficsigndetection.TSDNavigationAdapter.getItem (
    int position ) [inline]
```

This method queries the [TSDFragment](#) instance at a given position in the internal array which was delivered at construction. This is the same as the position of the menu item.

**Parameters**

<i>position</i>	Position of the menu item to get the <a href="#">TSDFragment</a> from.
-----------------	--

**Returns**

[TSDFragment](#) which is displayed at the given position as menu item.

**6.29.3.3 getItemId()**

```
long de.swp.tsd.trafficsigndetection.TSDNavigationAdapter.getItemId (
    int position ) [inline]
```

This method queries the id of the item at the given position. It just returns the position since the fragments itself should be unique in the list.

**Parameters**

<i>position</i>	Position of the menu item to get the id from.
-----------------	---

**Returns**

Id or position of the item at this position.

**6.29.3.4 getView()**

```
View de.swp.tsd.trafficsigndetection.TSDNavigationAdapter.getView (
    int position,
    View convertView,
    ViewGroup parent ) [inline]
```

This method updates a given view to display the menu item for the given position using the information given in the fragments array at this position.

**Parameters**

<i>position</i>	Position to display the menu item at.
<i>convertView</i>	View to display the information in.
<i>parent</i>	View parent the view is used in.

**Returns**

The menu item view with the updated information.

References `de.swp.tsd.trafficsigndetection.TSDFragment.displayIconId`, `de.swp.tsd.trafficsigndetection.TSDFragment.displayNameId`, and `de.swp.tsd.trafficsigndetection.TSDNavigationAdapter.getItem()`.

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDNavigationAdapter.java](#)

## 6.30 de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode Enum Reference

### Public Member Functions

- [TSDOpticalStabilizationMode](#) (int *value*)

### Static Public Member Functions

- static [TSDOpticalStabilizationMode fromInt](#) (int *value*)

### Public Attributes

- [OFF](#) =(0)
- [ON](#) =(1)
- final int *value*

### 6.30.1 Detailed Description

This enum wraps the possible modes of optical image stabilization of the camera instance according to the [Android API](#) into an enum. See the documentation for detailed information.

### 6.30.2 Constructor & Destructor Documentation

#### 6.30.2.1 TSDOpticalStabilizationMode()

```
de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode.TSDOpticalStabilizationMode (
    int value ) [inline]
```

This constructor creates a new enum instance setting the internal enum value to the given one.

#### Parameters

<i>value</i>	This value should match the value of the constant in the Android API describing the same mode.
--------------	--

### 6.30.3 Member Function Documentation

#### 6.30.3.1 fromInt()

```
static TSDOpticalStabilizationMode de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode.fromInt (
    int value ) [inline], [static]
```

This function can convert a given value which is one of the constant describing ois modes in the Android API to the corresponding enum value. If the given value does not match any defined enum, then OFF will be returned.

##### Parameters

<i>value</i>	Integer that should be converted.
--------------	-----------------------------------

##### Returns

Enum value that corresponds to the given int value. If not matching one is found, OFF will be returned.

### 6.30.4 Member Data Documentation

#### 6.30.4.1 OFF

```
de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode.OFF = (0)
```

The ois is not available. `LENS_OPTICAL_STABILIZATION_MODE_OFF`

#### 6.30.4.2 ON

```
de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode.ON = (1)
```

The ois is enabled. `LENS_OPTICAL_STABILIZATION_MODE_ON`

#### 6.30.4.3 value

```
final int de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode.value
```

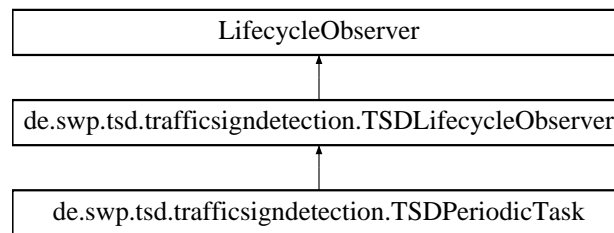
This value describes the value of the constant in the Android API describing the same mode this enum wraps.

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDOpticalStabilizationMode.java](#)

## 6.31 de.swp.tsd.trafficsigndetection.TSDPeriodicTask Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDPeriodicTask:



### Protected Member Functions

- void [onResume](#) ()
- void [onPause](#) ()

### Package Functions

- [TSDPeriodicTask](#) (long delayMillis, long periodMillis, @NonNull Runnable runnable, @NonNull Lifecycle lifecycle)

### Additional Inherited Members

#### 6.31.1 Detailed Description

This class defines an internal timer triggering the given runnable at a given interval. It is lifecycle aware using the [TSDLifecycleObserver](#) base class.

#### 6.31.2 Constructor & Destructor Documentation

##### 6.31.2.1 TSDPeriodicTask()

```

de.swp.tsd.trafficsigndetection.TSDPeriodicTask.TSDPeriodicTask (
    long delayMillis,
    long periodMillis,
    @NonNull Runnable runnable,
    @NonNull Lifecycle lifecycle ) [inline], [package]
  
```

This constructor creates a new instance of [TSDPeriodicTask](#). This does not enable the object. To enable the object please see [TSDLifecycleObserver#enable\(\)](#).

## Parameters

<i>delayMillis</i>	Delay after the runnable should get run the first time.
<i>periodMillis</i>	Interval the runnable should get run in in milliseconds.
<i>lifecycle</i>	Lifecycle of the parent starting and stopping the timer.

### 6.31.3 Member Function Documentation

#### 6.31.3.1 onPause()

```
void de.swp.tsd.trafficsigndetection.TSDPeriodicTask.onPause ( ) [inline], [protected]
```

This method gets called when the parent lifecycle hits the ON\_PAUSE event and the object is enabled. It cancels the timer and cleans it up.

#### 6.31.3.2 onResume()

```
void de.swp.tsd.trafficsigndetection.TSDPeriodicTask.onResume ( ) [inline], [protected]
```

This method gets called when the parent lifecycle hits the ON\_RESUME event and the object is enabled. It starts the timer when the object is enabled.

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPeriodicTask.java](#)

## 6.32 de.swp.tsd.trafficsigndetection.TSDPermissionManager Class Reference

### Classes

- interface [OnRequestPermissionResultListener](#)

### Static Public Member Functions

- static void [onRequestPermissionResult](#) (int requestCode, Activity activity, int[] grantResults, CoordinatorLayout view)
- static void [getPermissions](#) (Activity activity, [OnRequestPermissionResultListener](#) resultListener, long waitMillis)

### 6.32.1 Detailed Description

This class is only for static use. Its functions can be used for requesting app permissions on runtime and reacting on the users result. To use this class, a request handler method needs to be implemented in the request receiving activity like for example here: `MainActivity#onRequestPermissionsResult(int, String[], int[])`. Then permissions can be requested by calling `TSDPermissionManager#getPermissions(Activity, OnRequestPermissionsResultListener, long)`. The delivered listener will get notified, when the users decision is known. This could be even before asking the user for any permission if he has permitted them already or when he accepted all or declined at least one of them. The signaled result will be accordingly. When the user declined at least one of the permissions, a Snackbar will be displayed to him giving him information about the effects of the decline and a button to ask for the permissions again.

### 6.32.2 Member Function Documentation

#### 6.32.2.1 `getPermissions()`

```
static void de.swp.tsd.trafficsigndetection.TSDPermissionManager.getPermissions (
    Activity activity,
    OnRequestPermissionsResultListener resultListener,
    long waitMillis ) [inline], [static]
```

This function can be called to ask the user for runtime permissions like CAMERA.

#### Parameters

<i>activity</i>	The activity this is used in. Make sure to call <code>TSDPermissionManager.onRequestPermissionsResult</code> inside this activity from the <code>onRequestPermissionsResult</code> callback. In fragments you can get this with <code>getActivity()</code> , but make sure it is not null.
<i>resultListener</i>	Observer which gets signaled, when the users choice upon the requested permissions is known. It also contains the asked permissions and the message to show to the user when not all permissions were granted.
<i>waitMillis</i>	How many milliseconds to wait until requesting the permission. This can be used as safety guard because requesting permissions in <code>onResume</code> causes an infinite loop or just as convenience to not display the dialog immediately when a activity or fragment starts.

References `de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionsResultListener.getRequestedPermissions()`, and `de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionsResultListener.onRequestedPermissionsGranted()`.

#### 6.32.2.2 `onRequestPermissionsResult()`

```
static void de.swp.tsd.trafficsigndetection.TSDPermissionManager.onRequestPermissionsResult (
    int requestCode,
    Activity activity,
```

```
int [] grantResults,
CoordinatorLayout view ) [inline], [static]
```

This function is called from the main activity handling the request permission results. It signals the appropriate listener defined before when having called [TSDPermissionManager.getPermissions\(\)](#) which is identified by the requestCode. When at least one requested permission was declined, an info text and the possibility to reask for the permission will be displayed to the user as Snackbar.

#### Parameters

<i>requestCode</i>	Request code of the result used to find correct listener.
<i>activity</i>	Activity the request was processed in.
<i>grantResults</i>	Array of all results of the corresponding permissions
<i>view</i>	Coordinator view a snackbar with the in the listener defined decline text will be shown, when some of the requested permissions were declined.

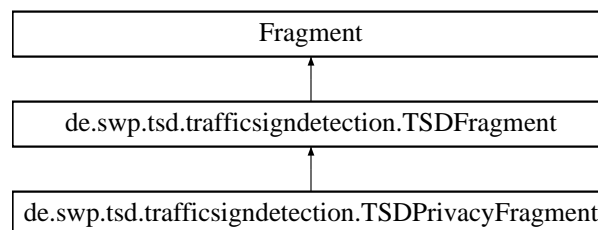
References [de.swp.tsd.trafficsigndetection.TSDPermissionManager.getPermissions\(\)](#), and [de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionsResultListener.onRequestedPermissionsGranted\(\)](#).

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPermissionManager.java](#)

## 6.33 de.swp.tsd.trafficsigndetection.TSDPrivacyFragment Class Reference

Inheritance diagram for [de.swp.tsd.trafficsigndetection.TSDPrivacyFragment](#):



#### Public Member Functions

- [TSDPrivacyFragment \(\)](#)
- void [onCreate](#) (Bundle savedInstanceState)
- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

#### Additional Inherited Members

##### 6.33.1 Detailed Description

This [TSDPrivacyFragment](#) is used to present the user the privacy policy of this application. It is the first [TSDPrivacyFragment](#) shown when the app is started the first time and will not give the possibility to continue to the main application without having it accepted.



## 6.33.2 Constructor & Destructor Documentation

### 6.33.2.1 TSDPrivacyFragment()

```
de.swp.tsd.trafficsigndetection.TSDPrivacyFragment.TSDPrivacyFragment ( ) [inline]
```

This constructor creates a new instance of [TSDPrivacyFragment](#) setting the correct values for [isRelevant name](#) and [isRelevant icon](#).

## 6.33.3 Member Function Documentation

### 6.33.3.1 onCreate()

```
void de.swp.tsd.trafficsigndetection.TSDPrivacyFragment.onCreate (
    Bundle savedInstanceState ) [inline]
```

This method gets called when the fragment is created. It initializes the fragment.

#### Parameters

<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.
---------------------------	--

### 6.33.3.2 onCreateView()

```
View de.swp.tsd.trafficsigndetection.TSDPrivacyFragment.onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState ) [inline]
```

This method gets called every time the view of the [TSDFragment](#) is newly created. It inflates the fragment\_license layout as user interface for the [TSDFragment](#).

#### See also

[TSDPrivacyFragment](#)

#### Parameters

<i>inflater</i>	Inflater to use for inflating the user interface layout of this fragment.
<i>container</i>	The place the ui should be inflated to, but not attached.
<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment.
Generated by Doxygen	Currently this does not get used.

**Returns**

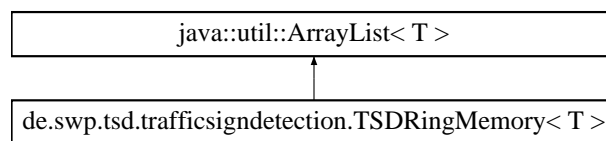
The inflated view containing the user interface of this [TSDFragment](#).

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDPrivacyFragment.java](#)

## 6.34 de.swp.tsd.trafficsigndetection.TSDRingMemory< T > Class Template Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDRingMemory< T >:

**Public Member Functions**

- boolean [add](#) (T item)
- void [add](#) (int index, T item)
- boolean [addAll](#) (Collection<? extends T > c)
- boolean [addAll](#) (int index, Collection<? extends T > c)
- [TSDRingMemory](#)< T > [getAllSkipLast](#) (int n, boolean reverse)
- void [setMaxSize](#) (int maxSize)
- int [getMaxSize](#) ()

**Package Functions**

- [TSDRingMemory](#) (int maxSize)

### 6.34.1 Detailed Description

Simple collection holding only a given maximum number of items.

**Parameters**

< T >	Item type
-------	-----------

### 6.34.2 Constructor & Destructor Documentation

### 6.34.2.1 TSDRingMemory()

```
de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.TSDRingMemory (
    int maxSize ) [inline], [package]
```

This constructor creates a new instance of [TSDRingMemory](#) setting the maximum number of items this instance should be able to hold.

#### Parameters

<i>maxSize</i>	Integer defining the maximum number of items in this collection.
----------------	--

## 6.34.3 Member Function Documentation

### 6.34.3.1 add() [1/2]

```
boolean de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.add (
    T item ) [inline]
```

This method adds a new item to the collection after having removed enough items at index 0 to make space for it.

#### Parameters

<i>item</i>	Item to add to collection.
-------------	----------------------------

#### Returns

Whether the collection changed as a result of this call.

### 6.34.3.2 add() [2/2]

```
void de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.add (
    int index,
    T item ) [inline]
```

This method adds a new item to the collection at the given position after having removed enough items at index 0 to make space for it.

#### Parameters

<i>index</i>	Index to add the item in.
<i>item</i>	Item to add to collection.

### 6.34.3.3 `addAll()` [1/2]

```
boolean de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.addAll (
    Collection<? extends T > c ) [inline]
```

This method adds the given collection to this collection after having removed enough items at index 0 to make space for them.

#### Parameters

<i>c</i>	Collection of items to add to this collection.
----------	--

#### Returns

Whether the collection changed as a result of this call.

### 6.34.3.4 `addAll()` [2/2]

```
boolean de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.addAll (
    int index,
    Collection<? extends T > c ) [inline]
```

This method adds the given collection to this collection at the specified index after having removed enough items at index 0 to make space for them.

#### Parameters

<i>index</i>	Index to add the items in.
<i>c</i>	Collection of items to add to this collection.

#### Returns

Whether the collection changed as a result of this call.

### 6.34.3.5 `getAllSkipLast()`

```
TSDRingMemory<T> de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.getAllSkipLast (
    int n,
    boolean reverse ) [inline]
```

This method creates a new instance of `TSDRingMemory` with all but the last *n* items in this instance. It can also reverse the order of the elements if wished. If reversed, still the last items in the original order will be skipped, not the first.

## Parameters

<i>n</i>	Number of last elements to skip.
<i>reverse</i>	Reverse the order of the remaining items.

## Returns

[TSDRingMemory](#) instance with all of the items of this instance but the last *n*.

## 6.34.3.6 getMaxSize()

```
int de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.getMaxSize ( ) [inline]
```

This method returns the maximum number of items this collection is able to hold.

## 6.34.3.7 setMaxSize()

```
void de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.setMaxSize (
    int maxSize ) [inline]
```

Set the maximum number of items this collection should be able to hold. When the new max size is smaller than the actual current size, then enough items at index 0 will get removed to match the new given max size.

## Parameters

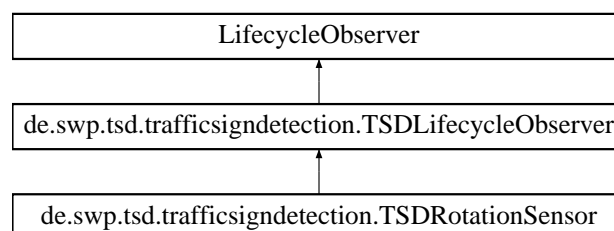
<i>maxSize</i>	Maximum number of items.
----------------	--------------------------

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDRingMemory.java](#)

## 6.35 de.swp.tsd.trafficsigndetection.TSDRotationSensor Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDRotationSensor:



## Classes

- interface [OnRotationListener](#)

## Protected Member Functions

- void [onResume](#) ()
- void [onPause](#) ()

## Package Functions

- [TSDRotationSensor](#) (@NonNull [OnRotationListener](#) rotationListener, @NonNull Lifecycle lifecycle, @NonNull Activity activity)

## Additional Inherited Members

### 6.35.1 Detailed Description

This class defines a lifecycle aware wrapper around the gravitation sensor calculating the device roll (rotation around the z axis) and pitch (rotation around the y axis), which get signaled every change to an [OnRotationListener](#).

### 6.35.2 Constructor & Destructor Documentation

#### 6.35.2.1 TSDRotationSensor()

```
de.swp.tsd.trafficsigndetection.TSDRotationSensor.TSDRotationSensor (
    @NonNull OnRotationListener rotationListener,
    @NonNull Lifecycle lifecycle,
    @NonNull Activity activity ) [inline], [package]
```

This constructor creates a new instance of [TSDRotationSensor](#). This does not enable the sensor listening. To enable the object please see [TSDLifecycleObserver#enable\(\)](#).

#### Parameters

<i>rotationListener</i>	Observer which should get notified on rotation events.
<i>lifecycle</i>	Lifecycle of the parent.
<i>activity</i>	Activity of the parent.

### 6.35.3 Member Function Documentation

## 6.35.3.1 onPause()

```
void de.swp.tsd.trafficsigndetection.TSDRotationSensor.onPause ( ) [inline], [protected]
```

This method gets called when the parent lifecycle hits the ON\_PAUSE event and the object is enabled. It stops the sensor listening.

## 6.35.3.2 onResume()

```
void de.swp.tsd.trafficsigndetection.TSDRotationSensor.onResume ( ) [inline], [protected]
```

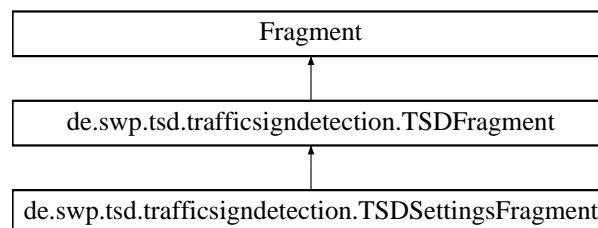
This method gets called when the parent lifecycle hits the ON\_RESUME event and the object is enabled. It starts the sensor listening.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDRotationSensor.java](#)

## 6.36 de.swp.tsd.trafficsigndetection.TSDSettingsFragment Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDSettingsFragment:



## Public Member Functions

- [TSDSettingsFragment](#) ()
- void [onCreate](#) (Bundle savedInstanceState)
- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

## Additional Inherited Members

## 6.36.1 Detailed Description

This [TSDFragment](#) enables the user to persistently set some settings to the app. Those settings are: Whether or not to use the speed tracking and the connected speed warning with a Switch. The threshold for the speed warning with a seek bar. The maximum number of sign combinations to hold in the sign history.

## 6.36.2 Constructor &amp; Destructor Documentation

### 6.36.2.1 TSDSettingsFragment()

```
de.swp.tsd.trafficsigndetection.TSDSettingsFragment.TSDSettingsFragment ( ) [inline]
```

This constructor creates a new instance of [TSDSettingsFragment](#) setting the correct values for [display name](#) and [display icon](#).

## 6.36.3 Member Function Documentation

### 6.36.3.1 onCreate()

```
void de.swp.tsd.trafficsigndetection.TSDSettingsFragment.onCreate (
    Bundle savedInstanceState ) [inline]
```

This method gets called when the fragment is created. It initializes the fragment.

#### Parameters

<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.
---------------------------	--

### 6.36.3.2 onCreateView()

```
View de.swp.tsd.trafficsigndetection.TSDSettingsFragment.onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState ) [inline]
```

This method gets called every time the view of the [TSDFragment](#) is newly created. It inflates the [fragment\\_ settings layout](#) as user interface for the [TSDFragment](#). It initializes the ranges for both seek bars (threshold and max number), sets the currently in user preferences saved settings to the views or default values and creates the listeners to updates the user preferences when the user changes some settings.

#### See also

[TSDSettingsFragment](#)

#### Parameters

<i>inflater</i>	Inflater to use for inflating the user interface layout of this fragment.
<i>container</i>	The place the ui should be inflated to, but not attached.
<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.



**Returns**

The inflated view containing the user interface of this [TSDFragment](#).

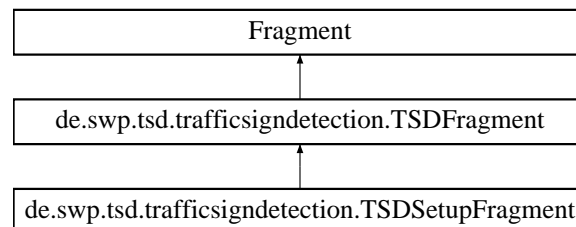
References `de.swp.tsd.trafficsigndetection.TSDHelper.readBooleanFromSharedPreferences()`, `de.swp.tsd.trafficsigndetection.TSDHelper.readIntegerFromSharedPreferences()`, `de.swp.tsd.trafficsigndetection.TSDHelper.writeBooleanToSharedPreferences()`, and `de.swp.tsd.trafficsigndetection.TSDHelper.writeIntegerToSharedPreferences()`.

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

- `App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSettingsFragment.java`

## 6.37 de.swp.tsd.trafficsigndetection.TSDSetupFragment Class Reference

Inheritance diagram for `de.swp.tsd.trafficsigndetection.TSDSetupFragment`:

**Public Member Functions**

- [TSDSetupFragment](#) ()
- void [onCreate](#) (Bundle savedInstanceState)
- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void [onStart](#) ()

**Additional Inherited Members**

### 6.37.1 Detailed Description

This [TSDFragment](#) defines an interactive manual to the user to setup his device physically correct. This includes hints for a power source, for the mounting and for the rotation using the device its gravity sensor. This three parts are separated into three different steps displayed in cards the user can scroll though. When the device is rotated correctly, then a green button is displayed with which the user pressing it can continue to the [detector screen](#).

### 6.37.2 Constructor & Destructor Documentation

### 6.37.2.1 TSDSetupFragment()

```
de.swp.tsd.trafficsigndetection.TSDSetupFragment.TSDSetupFragment ( ) [inline]
```

This constructor creates a new instance of [TSDSetupFragment](#) setting the correct values for icon and isRelevant name.

## 6.37.3 Member Function Documentation

### 6.37.3.1 onCreate()

```
void de.swp.tsd.trafficsigndetection.TSDSetupFragment.onCreate (
    Bundle savedInstanceState ) [inline]
```

This method gets called when the fragment is created. It initializes the fragment.

#### Parameters

<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.
---------------------------	--

### 6.37.3.2 onCreateView()

```
View de.swp.tsd.trafficsigndetection.TSDSetupFragment.onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState ) [inline]
```

This method gets called every time the view of the [TSDFragment](#) is newly created. It inflates the fragment\_setup layout as user interface for the [TSDFragment](#). It initializes the [TSDRotationSensor](#) to get updates about the current device rotation. On every update the rotation gets checked and the user gets displayed the correct instructions how to rotate the device to get it right. When one axis is rotated correctly, its degrees text says so and gets green. If all axis are rotated correctly, then the hint text itself also gets green and the button to continue gets displayed.

#### See also

[TSDSetupFragment](#)

#### Parameters

<i>inflater</i>	Inflater to use for inflating the user interface layout of this fragment.
<i>container</i>	The place the ui should be inflated to, but not attached.
<i>savedInstanceState</i>	Contains key-value pairs that might have been saved at the last destroy of the fragment. Currently this does not get used.

### Returns

The inflated view containing the user interface of this [TSDFragment](#).

References [de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.disable\(\)](#).

#### 6.37.3.3 onStart()

```
void de.swp.tsd.trafficsigndetection.TSDSetupFragment.onStart ( ) [inline]
```

References [de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.enable\(\)](#).

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSetupFragment.java](#)

## 6.38 de.swp.tsd.trafficsigndetection.TSDShutterCamera Class Reference

### Public Member Functions

- [TSDCameraCharacter](#) [getCameraCharacter](#) ()
- boolean [isEnabled](#) ()
- void [enable](#) ()
- void [disable](#) ()
- boolean [isInitiated](#) ()
- Integer [getSensorRotation](#) ()

### Package Functions

- [TSDShutterCamera](#) (@NonNull [TSDCameraManager](#) manager, @NonNull Lifecycle lifecycle, @NonNull Size imgTargetSize, TextureView preview, ImageReader.OnImageAvailableListener imageListener)

#### 6.38.1 Detailed Description

This class defines an behaviour for the [TSDBaseCamera](#) featuring the functionality to display a camera preview and take images which get send to the given [TSDShutterCameraImageListener](#). Exposure and white balance are set to auto while focus is set to infinite distance. Pictures are taken in [TSDCameraOutputFormat#YUV\\_420\\_888](#) format which the image listener is able to decode. The internal ImageReader is configured to hold a maximum of 2 images. The camera will fill those slots and only add a new picture, when at least one slot was processed by the [TSDShutterCameraImageListener](#).

### See also

[TSDShutterCameraImageListener](#)

## 6.38.2 Constructor & Destructor Documentation

### 6.38.2.1 TSDShutterCamera()

```
de.swp.tsd.trafficsigndetection.TSDShutterCamera.TSDShutterCamera (
    @NonNull TSDCameraManager manager,
    @NonNull Lifecycle lifecycle,
    @NonNull Size imgTargetSize,
    TextureView preview,
    ImageReader.OnImageAvailableListener imageListener ) [inline], [package]
```

This constructor creates a new instance of [TSDShutterCamera](#). Taking the manager it uses the best back facing camera available. If both preview and imageListener are null, no capture session will be started.

#### Parameters

<i>manager</i>	Camera manager the camera information are taken from.
<i>lifecycle</i>	Lifecycle of the parent used to manage the camera lifecycle.
<i>imgTargetSize</i>	Target size for the output images. The resulting images are at the most this big.
<i>preview</i>	TextureView the preview of the camera is displayed to. This might get removed at a later stage in development. If this is null, no preview will be used.
<i>imageListener</i>	Image observer getting signaled when an image from the camera is ready. If this is null, no image reader will be used.

References [de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getMatchingOutputSize\(\)](#).

## 6.38.3 Member Function Documentation

### 6.38.3.1 disable()

```
void de.swp.tsd.trafficsigndetection.TSDShutterCamera.disable ( ) [inline]
```

This method closes and disables the camera if not already the case.

References [de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.disable\(\)](#).

### 6.38.3.2 enable()

```
void de.swp.tsd.trafficsigndetection.TSDShutterCamera.enable ( ) [inline]
```

This method enables the camera if not already enabled. This does not necessarily open the camera, but it will at the latest when the parent lifecycle hits the ON\_RESUME event.

References [de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.enable\(\)](#).

### 6.38.3.3 getCameraCharacter()

```
TSDCameraCharacter de.swp.tsd.trafficsigndetection.TSDShutterCamera.getCameraCharacter ( )  
[inline]
```

References [de.swp.tsd.trafficsigndetection.TSDBaseCamera.getCharacter\(\)](#).

### 6.38.3.4 getSensorRotation()

```
Integer de.swp.tsd.trafficsigndetection.TSDShutterCamera.getSensorRotation ( ) [inline]
```

This method returns the value of [TSDCameraCharacter#getSensorRotation\(\)](#) of the internally selected camera for image capture.

#### Returns

Sensor rotation of selected camera or null, if camera was not initialized.

References [de.swp.tsd.trafficsigndetection.TSDLensFacing.BACK](#), [de.swp.tsd.trafficsigndetection.TSDBaseCamera.getBackgroundHandler\(\)](#), [de.swp.tsd.trafficsigndetection.TSDCameraManager.getCameraCharacters\(\)](#), [de.swp.tsd.trafficsigndetection.TSDBaseCamera.getCharacter\(\)](#), [de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getScore\(\)](#), and [de.swp.tsd.trafficsigndetection.TSDCameraCharacter.getSensorRotation\(\)](#).

### 6.38.3.5 isEnabled()

```
boolean de.swp.tsd.trafficsigndetection.TSDShutterCamera.isEnabled ( ) [inline]
```

Queries whether the camera is enabled or not. Enabled does not necessarily mean that it is running. Enabled just means that it will automatically start, when the parent lifecycle hits the onResume event and automatically stop on onPause.

#### Returns

Enable state of the camera.

References [de.swp.tsd.trafficsigndetection.TSDLifecycleObserver.isEnabled\(\)](#).

### 6.38.3.6 isInitiated()

```
boolean de.swp.tsd.trafficsigndetection.TSDShutterCamera.isInitiated ( ) [inline]
```

This method checks, whether the camera was created at construction or not.

#### Returns

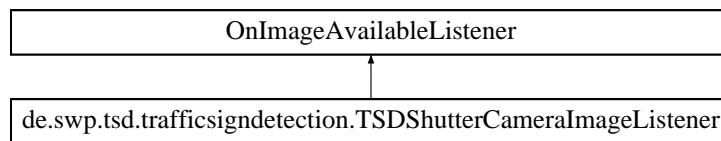
Boolean indication whether the camera was created or not.

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDShutterCamera.java](#)

## 6.39 de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener:



### Classes

- interface [FrameEventListener](#)

### Public Member Functions

- native int [ ][ ] [roadSignAPIfeedImage](#) (int width, int height, byte yuv[ ], int[ ] rgb, int stride, int rotate)
- void [setSensorRotation](#) (int sensorRotation)
- void [onImageAvailable](#) (final ImageReader reader)

### Package Functions

- [TSDShutterCameraImageListener](#) ([TSDSignHistory](#) history, [TSDDrawableDatabase](#) database, [ImageView](#) postView, [FrameEventListener](#) listener, @NonNull Activity activity)

### 6.39.1 Detailed Description

This class is used as image listener for [TSDShutterCamera](#) instances. It reads on every signal to [TSDShutterCameraImageListener#onImageAvailable\(ImageReader\)](#) the latest image in, extracts the yuv bytes and passes the data to the RoadSignAPI. The API returns the rgb image data and information about found signs, which get processed to [sign combinations](#) and are added to the given [TSDSignHistory](#). Currently only signs will be extracted from the image, when the yuv image delivered by the camera has a pixel stride of two.

### 6.39.2 Constructor & Destructor Documentation

#### 6.39.2.1 TSDShutterCameraImageListener()

```

de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.TSDShutterCameraImageListener (
    TSDSignHistory history,
    TSDDrawableDatabase database,
    ImageView postView,
    FrameEventListener listener,
    @NonNull Activity activity ) [inline], [package]
  
```

This constructor creates a new instance of [TSDShutterCameraImageListener](#). If the history or the database is null, then the sign combinations can not be added to some history.

## Parameters

<i>history</i>	Sign history the image listener should send the new recognized sign combis to.
<i>database</i>	Drawable database to be used for loading sing drawables.
<i>postView</i>	If not null, results from native are displayed in this image view.
<i>listener</i>	Listener which gets notified every time a frame is finished delivering information about the time it took to process it.
<i>activity</i>	Parent activity

References `de.swp.tsd.trafficsigndetection.TSDSignHistory.getLatestFrameId()`, and `de.swp.tsd.trafficsigndetection.TSDSignHistory.getLatestId()`.

### 6.39.3 Member Function Documentation

#### 6.39.3.1 onImageAvailable()

```
void de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.onImageAvailable (
    final ImageReader reader ) [inline]
```

This method gets signaled every time an image is available from the [TSDShutterCamera](#) instance this is used in. It reads in the latest image and processes it as described in [TSDShutterCameraImageListener](#). At the end the image is closed and thus the slot in the image reader freed for a new image from the camera.

## Parameters

<i>reader</i>	Image reader used
---------------	-------------------

References `de.swp.tsd.trafficsigndetection.TSDSignHistory.addSignCombination()`, `de.swp.tsd.trafficsigndetection.TSDSign.fromId()`, `de.swp.tsd.trafficsigndetection.TSDSignCombination.getDistanceToImageCenter()`, `de.swp.tsd.trafficsigndetection.TSDSign.isRelevant()`, `de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.FrameEventListener.onFrameFinished()`, and `de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.roadSignAPIfeedImage()`.

#### 6.39.3.2 roadSignAPIfeedImage()

```
native int [][] de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.roadSignAPIfeedImage (
    int width,
    int height,
    byte yuv[],
    int [] rgb,
    int stride,
    int rotate )
```

This method converts the given yuv image data with a pixel stride of two - which is most common, to rgb image data in native code. Then it feeds the image data into the RoadSign pipeline and extracts the found traffic signs.

## Parameters

<i>width</i>	Width of the image in pixels.
<i>height</i>	Height of the image in pixels.
<i>yuv</i>	Byte array containing the yuv data with pixel stride of two.
<i>rgb</i>	Byte array that will contain the rgb data after the method was called.
<i>stride</i>	Uv pixel stride of the yuv data to determine in which format the yuv byte array is written. If the stride is 1, then the yuv byte array is in I420 format, if the pixel stride is 2, then it is in NV21 format.
<i>rotate</i>	Integer to describe the necessary image rotation to get straight up.

## Returns

2D array of integers containing information about the found signs grouped in sign combinations. Each array in this array defines one combination of signs. The first value in each array contains the number of signs in the combination. Then following for each sign in this order there is the classification id of the detected sign, the x and y axis of the upper left corner of the rect where the sign was detected in the image and the x and y axis of the lower right corner of the detected rect.

## 6.39.3.3 setSensorRotation()

```
void de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.setSensorRotation (
    int sensorRotation ) [inline]
```

Setter for the sensor rotation value used to correct image orientation problems. This value should be the one queried from [TSDCameraCharacter#getSensorRotation\(\)](#).

## Parameters

<i>sensorRotation</i>	<a href="#">TSDCameraCharacter#getSensorRotation()</a> of the camera used to capture the images.
-----------------------	--

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDShutterCameraImageListener.java](#)

## 6.40 de.swp.tsd.trafficsigndetection.TSDSign Enum Reference

## Public Member Functions

- [TSDSign](#) (int speedLimit)
- boolean [isZoneStart](#) ()
- boolean [endsZone](#) ([TSDSign](#) zoneStart)
- boolean [isRelevant](#) ()
- boolean [isStandalone](#) ()
- [TSDDrawable](#) [getDrawable](#) (@NonNull [TSDDrawableDatabase](#) database, @NonNull Context context)



## Static Public Member Functions

- static [TSDSign fromId](#) (int id)

## Public Attributes

- [MISC](#) =(0)
- [VZ523\\_30](#) =(0)
- [MISC\\_BLUE](#) =(0)
- [MISC\\_RED\\_BLUE](#) =(0)
- [MISC\\_RED\\_WHITE](#) =(0)
- [MISC\\_WHITE](#) =(0)
- [MISC\\_YELLOW](#) =(0)
- [MISC\\_BROWN](#) =(0)
- [MISC\\_GREEN](#) =(0)
- [VZ1000](#) =(0)
- [VZ1001\\_30](#) =(0)
- [VZ1001\\_31](#) =(0)
- [VZ1001\\_32](#) =(0)
- [VZ1001\\_33](#) =(0)
- [VZ1004\\_30](#) =(0)
- [VZ1004\\_31](#) =(0)
- [VZ101](#) =(0)
- [VZ101\\_11](#) =(0)
- [VZ103\\_10](#) =(0)
- [VZ103\\_20](#) =(0)
- [VZ1040\\_30](#) =(0)
- [VZ1042\\_31](#) =(0)
- [VZ1042\\_33](#) =(0)
- [VZ105\\_10](#) =(0)
- [VZ105\\_20](#) =(0)
- [VZ1053\\_35](#) =(0)
- [VZ120](#) =(0)
- [VZ121\\_10](#) =(0)
- [VZ121\\_20](#) =(0)
- [VZ123](#) =(0)
- [VZ131](#) =(0)
- [VZ133\\_10](#) =(0)
- [VZ136\\_10](#) =(0)
- [VZ274\\_10](#) =(10)
- [VZ274\\_20](#) =(20)
- [VZ274\\_30](#) =(30)
- [VZ274\\_40](#) =(40)
- [VZ274\\_50](#) =(50)
- [VZ274\\_60](#) =(60)
- [VZ274\\_70](#) =(70)
- [VZ274\\_80](#) =(80)
- [VZ274\\_90](#) =(90)
- [VZ274\\_100](#) =(100)
- [VZ274\\_110](#) =(110)
- [VZ274\\_120](#) =(120)
- [VZ274\\_130](#) =(130)
- [VZ274x1](#) =(30)
- [VZ274x2](#) =(-1)

- [VZ278](#) =(-1)
- [VZ282](#) =(-1)
- [VZ306](#) =(0)
- [VZ310](#) =(50)
- [VZ311](#) =(-1)
- [VZ325x1](#) =(10)
- [VZ325x2](#) =(-1)
- [VZ330x1](#) =(1\_000\_000)
- [VZ330x2](#) =(-1)
- [VZ331x1](#) =(100)
- [VZ331x2](#) =(-1)
- final int [speedLimit](#)

### 6.40.1 Detailed Description

This enum defines (more than) the traffic signs, that can be detected by the RoadSignAPI. The enums hold a value describing their effect on the current speed limit. The names of the enums correspond to in most cases their full official id, and in some to the starting part of it. Here an "\_" stands for "-" and an "x" for an "." in the official id. Only MISC\* are not official ids and used for every sign that could not be classified to one of the other signs. VZ278 is an broader id for all sub ids.

### 6.40.2 Constructor & Destructor Documentation

#### 6.40.2.1 TSDSign()

```
de.swp.tsd.trafficsigndetection.TSDSign.TSDSign (
    int speedLimit ) [inline]
```

This constructor creates a new instance of [TSDSign](#).

#### Parameters

<i>speedLimit</i>	This value describes the direct effect of the sign on the allowed speed. =0 means no direct effect or no effect that we could handle properly. >0 means specific speed limit. <0 means annulment of the last speed limit, but not necessarily no speed limit.
-------------------	---

### 6.40.3 Member Function Documentation

#### 6.40.3.1 endsZone()

```
boolean de.swp.tsd.trafficsigndetection.TSDSign.endsZone (
    TSDSign zoneStart ) [inline]
```

This method checks, whether this sign ends the special speed zone started by the given sign. Mind though, that zones can be nested. For example taking the incoming signs { town start, 30 start, 30 end, town end }, the town zone is only ended with the last sign, not with the start of the 30 zone. This is, because after the end of the 30 zone the town zone is still relevant. A higher level speed zone end will also end every lower level speed zone end, where higher and lower level directly refers the the allowed speed in the zone.

#### Parameters

<i>zoneStart</i>	Sign which started the special speed zone (can be detected by <a href="#">TSDSign#isZoneStart()</a> ).
------------------	--

#### Returns

Whether this sign ends the zone or not.

#### 6.40.3.2 fromId()

```
static TSDSign de.swp.tsd.trafficsigndetection.TSDSign.fromId(
    int id ) [inline], [static]
```

This function delivers the correct sign enum for the given classification class id returned by the RoadSignAPI.

#### Parameters

<i>id</i>	Classification class id from RoadSignAPI.
-----------	---

#### Returns

Matching sign enum for classification class id or if not found MISC.

#### 6.40.3.3 getDrawable()

```
TSDDrawable de.swp.tsd.trafficsigndetection.TSDSign.getDrawable (
    @NonNull TSDDrawableDatabase database,
    @NonNull Context context ) [inline]
```

This method tries to load the matching drawable from resources to the given sign enum instance. For every sign not present in the resources null will be returned.

#### Parameters

<i>database</i>	Database to use for drawable loading.
<i>context</i>	Context the sign should be loaded in.

**Returns**

The found drawable or null.

References `de.swp.tsd.trafficsigndetection.TSDDrawable.getDrawable()`.

**6.40.3.4 isRelevant()**

```
boolean de.swp.tsd.trafficsigndetection.TSDSign.isRelevant ( ) [inline]
```

This method queries, whether this sign should be saved into the history or not. This is used for recognized but irrelevant signs to exclude them from history and displaying to the user.

**Returns**

Whether this sign should be included in the history or not.

**6.40.3.5 isStandalone()**

```
boolean de.swp.tsd.trafficsigndetection.TSDSign.isStandalone ( ) [inline]
```

This method returns, whether this sign is appropriate to be displayed alone in one combination or if in this case, that the combination only consists of this or multiple signs with the same result, the sign combination should not get displayed to the user. If one combination consists of standalone and not standalone signs, then the combination gets displayed.

**Returns**

Whether the sign can stand alone or not.

**6.40.3.6 isZoneStart()**

```
boolean de.swp.tsd.trafficsigndetection.TSDSign.isZoneStart ( ) [inline]
```

This method checks whether this sign is the start of a special speed zone. Those zones can be a calmed zone, a 30 zone, a town zone, an automotive road and an autobahn.

**Returns**

Whether this sign is the start of a special speed zone.

**6.40.4 Member Data Documentation**

#### 6.40.4.1 MISC

```
de.swp.tsd.trafficsigndetection.TSDSign.MISC =(0)
```

#### 6.40.4.2 MISC\_BLUE

```
de.swp.tsd.trafficsigndetection.TSDSign.MISC_BLUE =(0)
```

#### 6.40.4.3 MISC\_BROWN

```
de.swp.tsd.trafficsigndetection.TSDSign.MISC_BROWN =(0)
```

#### 6.40.4.4 MISC\_GREEN

```
de.swp.tsd.trafficsigndetection.TSDSign.MISC_GREEN =(0)
```

#### 6.40.4.5 MISC\_RED\_BLUE

```
de.swp.tsd.trafficsigndetection.TSDSign.MISC_RED_BLUE =(0)
```

#### 6.40.4.6 MISC\_RED\_WHITE

```
de.swp.tsd.trafficsigndetection.TSDSign.MISC_RED_WHITE =(0)
```

#### 6.40.4.7 MISC\_WHITE

```
de.swp.tsd.trafficsigndetection.TSDSign.MISC_WHITE =(0)
```

#### 6.40.4.8 MISC\_YELLOW

```
de.swp.tsd.trafficsigndetection.TSDSign.MISC_YELLOW =(0)
```

#### 6.40.4.9 speedLimit

```
final int de.swp.tsd.trafficsigndetection.TSDSign.speedLimit
```

This value describes the direct effect of the sign on the allowed speed. =0 means no direct effect or no effect that could handled properly (e.g. limits which are limited in range) >0 means specific speed limit. <0 means annulment of the last speed limit, but not necessarily no speed limit.

#### 6.40.4.10 VZ1000

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1000 =(0)
```

#### 6.40.4.11 VZ1001\_30

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1001_30 =(0)
```

#### 6.40.4.12 VZ1001\_31

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1001_31 =(0)
```

#### 6.40.4.13 VZ1001\_32

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1001_32 =(0)
```

#### 6.40.4.14 VZ1001\_33

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1001_33 =(0)
```

#### 6.40.4.15 VZ1004\_30

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1004_30 =(0)
```

**6.40.4.16 VZ1004\_31**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1004_31 =(0)
```

**6.40.4.17 VZ101**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ101 =(0)
```

**6.40.4.18 VZ101\_11**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ101_11 =(0)
```

**6.40.4.19 VZ103\_10**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ103_10 =(0)
```

**6.40.4.20 VZ103\_20**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ103_20 =(0)
```

**6.40.4.21 VZ1040\_30**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1040_30 =(0)
```

**6.40.4.22 VZ1042\_31**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1042_31 =(0)
```

**6.40.4.23 VZ1042\_33**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1042_33 =(0)
```

**6.40.4.24 VZ1053\_35**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ1053_35 =(0)
```

**6.40.4.25 VZ105\_10**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ105_10 =(0)
```

**6.40.4.26 VZ105\_20**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ105_20 =(0)
```

**6.40.4.27 VZ120**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ120 =(0)
```

**6.40.4.28 VZ121\_10**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ121_10 =(0)
```

**6.40.4.29 VZ121\_20**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ121_20 =(0)
```

**6.40.4.30 VZ123**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ123 =(0)
```

**6.40.4.31 VZ131**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ131 =(0)
```



**6.40.4.32 VZ133\_10**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ133_10 =(0)
```

**6.40.4.33 VZ136\_10**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ136_10 =(0)
```

**6.40.4.34 VZ274\_10**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_10 =(10)
```

**6.40.4.35 VZ274\_100**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_100 =(100)
```

**6.40.4.36 VZ274\_110**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_110 =(110)
```

**6.40.4.37 VZ274\_120**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_120 =(120)
```

**6.40.4.38 VZ274\_130**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_130 =(130)
```

**6.40.4.39 VZ274\_20**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_20 =(20)
```

**6.40.4.40 VZ274\_30**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_30 =(30)
```

**6.40.4.41 VZ274\_40**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_40 =(40)
```

**6.40.4.42 VZ274\_50**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_50 =(50)
```

**6.40.4.43 VZ274\_60**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_60 =(60)
```

**6.40.4.44 VZ274\_70**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_70 =(70)
```

**6.40.4.45 VZ274\_80**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_80 =(80)
```

**6.40.4.46 VZ274\_90**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274_90 =(90)
```

**6.40.4.47 VZ274x1**

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ274x1 =(30)
```

**6.40.4.48 VZ274x2**

de.swp.tsd.trafficsigndetection.TSDSign.VZ274x2 =(-1)

**6.40.4.49 VZ278**

de.swp.tsd.trafficsigndetection.TSDSign.VZ278 =(-1)

**6.40.4.50 VZ282**

de.swp.tsd.trafficsigndetection.TSDSign.VZ282 =(-1)

**6.40.4.51 VZ306**

de.swp.tsd.trafficsigndetection.TSDSign.VZ306 =(0)

**6.40.4.52 VZ310**

de.swp.tsd.trafficsigndetection.TSDSign.VZ310 =(50)

**6.40.4.53 VZ311**

de.swp.tsd.trafficsigndetection.TSDSign.VZ311 =(-1)

**6.40.4.54 VZ325x1**

de.swp.tsd.trafficsigndetection.TSDSign.VZ325x1 =(10)

**6.40.4.55 VZ325x2**

de.swp.tsd.trafficsigndetection.TSDSign.VZ325x2 =(-1)

#### 6.40.4.56 VZ330x1

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ330x1 =(1_000_000)
```

#### 6.40.4.57 VZ330x2

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ330x2 =(-1)
```

#### 6.40.4.58 VZ331x1

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ331x1 =(100)
```

#### 6.40.4.59 VZ331x2

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ331x2 =(-1)
```

#### 6.40.4.60 VZ523\_30

```
de.swp.tsd.trafficsigndetection.TSDSign.VZ523_30 =(0)
```

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDSign.java](#)

## 6.41 de.swp.tsd.trafficsigndetection.TSDSignCombination Class Reference

### Public Member Functions

- long [getFrameld](#) ()
- long [getId](#) ()
- int [getDistanceToImageCenter](#) ()
- [TSDDrawable](#) [] [getSignDrawables](#) ()
- [TSDSign](#) [] [getSigns](#) ()
- long [getTimestamp](#) ()
- void [setSpeed](#) (Integer speed)
- Integer [getSpeed](#) ()
- boolean [isStandalone](#) ()
- boolean [isSublistOf](#) ([TSDSignCombination](#) c)
- [TSDSignCombination](#) [createUpdatedBySublist](#) ([TSDSignCombination](#) newCombi)
- boolean [equals](#) (Object obj)

## Package Functions

- [TSDSignCombination](#) (long id, long frameId, @NonNull [TSDSign](#)[] signs, @NonNull Rect[] rects, Bitmap image, long timestamp, Integer speed, @NonNull [TSDDrawableDatabase](#) database, @NonNull Context context)

### 6.41.1 Detailed Description

This class holds all information about the combination of multiple or one traffic signs and additionally the time and speed. This includes one id which should be unique for one pole of traffic signs in real life. For the definition of the signs there an array of [sign enums](#) describing their classes and an array of [sign drawables](#) describing their images which can be displayed to the user. For time isRelevant just the timestamp in milliseconds when the combination was recognized is saved.

### 6.41.2 Constructor & Destructor Documentation

#### 6.41.2.1 TSDSignCombination()

```
de.swp.tsd.trafficsigndetection.TSDSignCombination.TSDSignCombination (
    long id,
    long frameId,
    @NonNull TSDSign [] signs,
    @NonNull Rect [] rects,
    Bitmap image,
    long timestamp,
    Integer speed,
    @NonNull TSDDrawableDatabase database,
    @NonNull Context context ) [inline], [package]
```

This constructor creates a new instance of [TSDSignCombination](#) loading the drawables by their given ids from the database. If the database cannot deliver a drawable, then the defined cutout will be taken as drawable from the given image for this image. This way it is ensured that every sign, that is returned by the RoadSignAPI, can also be displayed to the user.

#### Parameters

<i>id</i>	Id of this sign combination. This can be used for replacing wrong recognition.
<i>signs</i>	Enum for each sign drawables in order top to bottom. Each sign enum must correspond to the image cutout rect in rects at the same index.
<i>rects</i>	Image cutout rect for each sign in order top to bottom. Each rect must correspond to the enum in signs at the same index.
<i>frameId</i>	Id describing the number of the frame in which this sign combination was captured. The frame id should be continuously increased by one for each frame that is analysed.
<i>image</i>	Image the rects define cutouts in.
<i>timestamp</i>	Time in milliseconds when the signs were recognized.
<i>speed</i>	Current speed or speed when the sign was recognized.
<i>database</i>	Database for drawables for more efficient loading.
<i>context</i>	Context for loading from resources.

References `de.swp.tsd.trafficsigndetection.TSDDrawable.getDrawable()`, and `de.swp.tsd.trafficsigndetection.TSD↔Drawable.getWidth()`.

### 6.41.3 Member Function Documentation

#### 6.41.3.1 `createUpdatedBySublist()`

```
TDSignCombination de.swp.tsd.trafficsigndetection.TDSignCombination.createUpdatedBySublist (
    TDSignCombination newCombi ) [inline]
```

This method will create a new instance of [TDSignCombination](#), if the given `combi` is a sublist of this instance. The signs will be copied over from this instance while getting updated by the same signs in the new combination. All other information will get copied over from the new instance.

##### Parameters

<i>newCombi</i>	The sign combination which information should be a update.
-----------------	--

##### Returns

When the given `combi` was a sublist, a new updated `combi` will be returned. Otherwise null will be returned.

References `de.swp.tsd.trafficsigndetection.TDSignCombination.equals()`, `de.swp.tsd.trafficsigndetection.TSD↔SignCombination.isSublistOf()`, and `de.swp.tsd.trafficsigndetection.TDSignCombination.TDSignCombination()`.

#### 6.41.3.2 `equals()`

```
boolean de.swp.tsd.trafficsigndetection.TDSignCombination.equals (
    Object obj ) [inline]
```

This method can be used to compare [TDSignCombination](#) instances. Those two instances match, if the following information are equal: the timestamp, the speed, the array of sign drawables and the frame id.

##### Parameters

<i>obj</i>	Other <a href="#">TDSignCombination</a> to compare to.
------------	--

##### Returns

The result of the comparison or false, if the given object to compare to was not of the same class as this instance.

References `de.swp.tsd.trafficsigndetection.TSDHelper.equalNullable()`, `de.swp.tsd.trafficsigndetection.TS↔DDrawable.getDrawable()`, `de.swp.tsd.trafficsigndetection.TSDDrawable.getDrawableId()`, and `de.swp.tsd.↔trafficsigndetection.TDSignCombination.TDSignCombination()`.

#### 6.41.3.3 getDistanceToImageCenter()

```
int de.swp.tsd.trafficsigndetection.TSDSignCombination.getDistanceToImageCenter ( ) [inline]
```

Getter for the average horizontal distance of this sign combination from the image center they were detected in. The distance is calculated between the horizontal center of the image and the average horizontal centers of the given sign combination rects.

##### Returns

Average horizontal distance of this sign combination from the image center. If the at constructions given image was null, then this will return 0;

#### 6.41.3.4 getFrameId()

```
long de.swp.tsd.trafficsigndetection.TSDSignCombination.getFrameId ( ) [inline]
```

Getter for the id of the frame this sign combination was captured in. The frame ids are increased by one for every analysed frame. So frame id 4 means one frame later than frame id 3.

##### Returns

Id of the frame the image was captured in.

#### 6.41.3.5 getId()

```
long de.swp.tsd.trafficsigndetection.TSDSignCombination.getId ( ) [inline]
```

Getter for the id of the sign combination. The same id does not mean that the content is the same, too. One id describes one sign combination in real life. If the recognition was wrong and some signs gets updated, then the id is still the same.

##### Returns

Id of this sign combination instance.

#### 6.41.3.6 getSignDrawables()

```
TSDDrawable [] de.swp.tsd.trafficsigndetection.TSDSignCombination.getSignDrawables ( ) [inline]
```

Getter for the contained sign drawables.

##### Returns

Array of drawables which are the signs.

#### 6.41.3.7 getSigns()

```
TSDSign [] de.swp.tsd.trafficsigndetection.TSDSignCombination.getSigns ( ) [inline]
```

Getter for the contained sign enums.

##### Returns

Array of enums which define the sign classes.

#### 6.41.3.8 getSpeed()

```
Integer de.swp.tsd.trafficsigndetection.TSDSignCombination.getSpeed ( ) [inline]
```

Getter for the current speed or the speed when the sign was recognized.

##### Returns

Speed

#### 6.41.3.9 getTimestamp()

```
long de.swp.tsd.trafficsigndetection.TSDSignCombination.getTimestamp ( ) [inline]
```

Getter for the timestamp the signs were recognized.

##### Returns

Timestamp the signs were recognized.



## 6.41.3.10 isStandalone()

```
boolean de.swp.tsd.trafficsigndetection.TSDSignCombination.isStandalone ( ) [inline]
```

This method checks, whether this combination contains at least one sign, that is standalone according to [TSDSign#isStandalone\(\)](#). Then the combination is standalone. If a sign combination is not standalone, then it will not be displayed to the user.

## Returns

Whether this sign combination is standalone or not.

## 6.41.3.11 isSublistOf()

```
boolean de.swp.tsd.trafficsigndetection.TSDSignCombination.isSublistOf (
    TSDSignCombination c ) [inline]
```

This method tests whether the signs of this combination are all contained in a given sign combination in this order. The given sign combination can contain more signs, it just needs to contain the signs in the correct order with others in between allowed. This only tests for sign types, not drawables.

## Parameters

<i>c</i>	Sign combination to test, whether it contains the signs in this order.
----------	--

## Returns

Whether the given sign combination contains the signs of this combination in this order.

## 6.41.3.12 setSpeed()

```
void de.swp.tsd.trafficsigndetection.TSDSignCombination.setSpeed (
    Integer speed ) [inline]
```

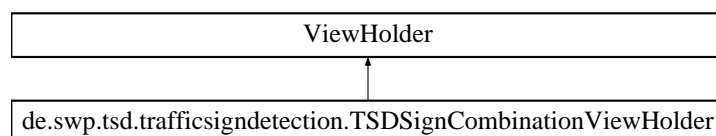
Setter for the current speed or the speed when the sign was recognized. This can also be null.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDSignCombination.java](#)

## 6.42 de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder:



## Public Member Functions

- void [setLeftRightMargin](#) (int leftDp, int rightDp)
- void [displayAllChanges](#) ([TSDSignCombination](#) oldCombi, [TSDSignCombination](#) newCombi, float sign←→ Transparency, float warningTransparency)
- void [displayAll](#) ([TSDSignCombination](#) signs, float signTransparency, float warningTransparency)
- void [setWarningTransparency](#) (float warningTransparency)
- void [displaySpeed](#) (Integer speed)
- void [displayTime](#) (long timestamp)
- void [displaySigns](#) ([TSDDrawable\[\]](#) signs, float transparency)

## Package Functions

- [TSDSignCombinationViewHolder](#) (View view)

### 6.42.1 Detailed Description

This class defines an recycler view holder, that can be used to display a [TSDSignCombination](#) to a sign card defined by the sign\_card.xml layout. In this layout the signs are displayed one below the one before like a pole of signs would look in real life. Furthermore the timespan since this combination was recognized is displayed in steps (0s, 1s and then every 5s, then every minute, then every hour and at last every day). On the sign card also speed can be displayed. For the display of the speed warning it is possible to set the alpha value of a red border around the card. For differentiation between the latest sign combination and the history it is possible to set the alpha value of the Signs.

### 6.42.2 Constructor & Destructor Documentation

#### 6.42.2.1 TSDSignCombinationViewHolder()

```
de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.TSDSignCombinationViewHolder (
    View view )    [inline], [package]
```

This constructor creates a new instance of [TSDSignCombinationViewHolder](#), searching for the necessary child views inside the given one. Therefore the given view must be a view inflated from sign\_card.xml.

#### Parameters

<i>view</i>	Sign card view inflated from sign_card.xml.
-------------	---

### 6.42.3 Member Function Documentation

## 6.42.3.1 displayAll()

```
void de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displayAll (
    TSDSignCombination signs,
    float signTransparency,
    float warningTransparency ) [inline]
```

This method displays every information given in the sign combination to the sign card view given to this instance.

## Parameters

<i>signs</i>	Sign combination to display.
<i>signTransparency</i>	Transparency of the image views showing the signs. This should be a value ranging from 0 to 1.
<i>warningTransparency</i>	Transparency of the warning border around the sign. This should be a value ranging from 0 to 1.

References de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displaySigns(), de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displaySpeed(), de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displayTime(), de.swp.tsd.trafficsigndetection.TSDSignCombination.getSignDrawables(), de.swp.tsd.trafficsigndetection.TSDSignCombination.getSpeed(), de.swp.tsd.trafficsigndetection.TSDSignCombination.getTimestamp(), and de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.setWarningTransparency().

## 6.42.3.2 displayAllChanges()

```
void de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displayAllChanges (
    TSDSignCombination oldCombi,
    TSDSignCombination newCombi,
    float signTransparency,
    float warningTransparency ) [inline]
```

This method displays every given information from the new combi which has changed from the given old combi. The time will always be refreshed, since its displayed as range to the current time.

## Parameters

<i>oldCombi</i>	The sign combi which was displayed here before.
<i>newCombi</i>	The sign combi which should be displayed.
<i>signTransparency</i>	Transparency of the image views showing the signs.
<i>warningTransparency</i>	Transparency of the warning border around the sign.

References de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displaySigns(), de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displaySpeed(), de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displayTime(), de.swp.tsd.trafficsigndetection.TSDHelper.equalNullable(), de.swp.tsd.trafficsigndetection.TSDSignCombination.getFrameld(), de.swp.tsd.trafficsigndetection.TSDSignCombination.getSignDrawables(), de.swp.tsd.trafficsigndetection.TSDSignCombination.getSpeed(), de.swp.tsd.trafficsigndetection.TSDSignCombination.getTimestamp(), and de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.setWarningTransparency().

### 6.42.3.3 displaySigns()

```
void de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displaySigns (
    TSDDrawable [] signs,
    float transparency ) [inline]
```

This method displays the given sign drawables to the connected sign card view in the given order from top to bottom like a pole with traffic signs in real life. This automatically scales everything so it fits best into the available space on the card.

#### Parameters

<i>signs</i>	The sign drawables to display.
<i>transparency</i>	The transparency of the image views showing the signs. This should be a value ranging from 0 to 1.

References `de.swp.tsd.trafficsigndetection.TSDHelper.dpToPx()`, `de.swp.tsd.trafficsigndetection.TSDDrawable.getDrawable()`, `de.swp.tsd.trafficsigndetection.TSDDrawable.getHeight()`, and `de.swp.tsd.trafficsigndetection.TSDDrawable.getWidth()`.

### 6.42.3.4 displaySpeed()

```
void de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displaySpeed (
    Integer speed ) [inline]
```

This method displays some given speed to the connected speed card view.

#### Parameters

<i>speed</i>	Speed to display. If this is null, then the speed view will be hidden. If the value is negative, then the indicator ~ will be displayed.
--------------	--

### 6.42.3.5 displayTime()

```
void de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.displayTime (
    long timestamp ) [inline]
```

This method displays the timespan from now to the given timestamp to the connected sign card view in steps (0s, 1s and then every 5s, then every minute, then every hour and at last every day).

#### Parameters

<i>timestamp</i>	Timestamp in milliseconds.
------------------	----------------------------

## 6.42.3.6 setLeftRightMargin()

```
void de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.setLeftRightMargin (
    int leftDp,
    int rightDp ) [inline]
```

This method sets the left and right margin of the sign card view in dp.

## Parameters

<i>leftDp</i>	Value in dp to set for left margin.
<i>rightDp</i>	Value in dp to set for right margin.

References `de.swp.tsd.trafficsigndetection.TSDHelper.dpToPx()`.

## 6.42.3.7 setWarningTransparency()

```
void de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder.setWarningTransparency (
    float warningTransparency ) [inline]
```

This method sets the transparency of the warning border of the sign card.

## Parameters

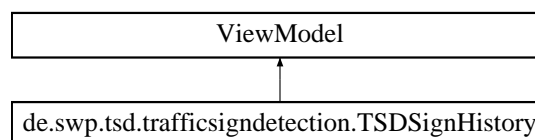
<i>warningTransparency</i>	Transparency of the warning border around the sign. This should be a value ranging from 0 to 1.
----------------------------	---

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

- `App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignCombinationViewHolder.java`

## 6.43 de.swp.tsd.trafficsigndetection.TSDSignHistory Class Reference

Inheritance diagram for `de.swp.tsd.trafficsigndetection.TSDSignHistory`:



## Public Member Functions

- `TSDSignHistory ()`
- `MutableLiveData< TSDRingMemory< TSDSignCombination > > getSignCombinations ()`
- `long getLatestFrameId ()`
- `long getLatestId ()`
- `void addSignCombination (TSDSignCombination signCombination, Activity activity)`
- `boolean isSpeedOk (Integer speed, int threshold)`

### 6.43.1 Detailed Description

This class acts as a collection for the recognized sign combinations. This class extends `ViewModel`. This way it is possible to maintain the history for the lifetime of the `MainActivity` even when it is created inside a `TSDFragment`. It is just important to not directly use the constructor but to use the builder factory like this: `ViewModelProviders.of(mActivity).get(TSDSignHistory.class)` Furthermore the `sign combinations` are stored inside `MutableLiveData` what makes it possible to register as an listener to and to receive signals whenever the collection is changed. The history also contains a `TSDSpeedChecker` to track the currently allowed speed limit using the history of recognized signs.

### 6.43.2 Constructor & Destructor Documentation

#### 6.43.2.1 TSDSignHistory()

```
de.swp.tsd.trafficsigndetection.TSDSignHistory.TSDSignHistory ( ) [inline]
```

This constructor creates a new instance of `TSDHistorySign`. Please do not call this directly but use the builder factory like this: `ViewModelProviders.of(mActivity).get(TSDSignHistory.class);`

### 6.43.3 Member Function Documentation

#### 6.43.3.1 addSignCombination()

```
void de.swp.tsd.trafficsigndetection.TSDSignHistory.addSignCombination (
    TSDSignCombination signCombination,
    Activity activity ) [inline]
```

This method adds a new sign combination to the back of the internal collection and signals all listener to the sign live data. It also updates the `TSDSignHistory#mSpeedChecker` In case the given sign combination is found to be a match to another combi in the history, then they will can be merged into an updated combination that is added instead of the given one. If the given sign combi is not standalone according to `TSDSignCombination#isStandalone()` then it will at most be used for updating purposes and then get discarded because non-standalone combis do not get displayed to the user and once added to the history they cannot become standalone through updates.

#### Parameters

<i>signCombination</i>	Sign combination to add to collection.
------------------------	--

References `de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.add()`, `de.swp.tsd.trafficsigndetection.TSDSignCombination.getId()`, `de.swp.tsd.trafficsigndetection.TSDSignCombination.getSigns()`, and `de.swp.tsd.trafficsigndetection.TSDSignCombination.isStandalone()`.

#### 6.43.3.2 getLatestFrameId()

```
long de.swp.tsd.trafficsigndetection.TSDSignHistory.getLatestFrameId ( ) [inline]
```

This method searches the history for the latest frame id. This can be used to use the correct start value when relaunching or starting the detection.

##### Returns

Latest frame id.

#### 6.43.3.3 getLatestId()

```
long de.swp.tsd.trafficsigndetection.TSDSignHistory.getLatestId ( ) [inline]
```

This method searches the history for the latest id. This can be used to use the correct start value when relaunching or starting the detection.

##### Returns

Latest id.

#### 6.43.3.4 getSignCombinations()

```
MutableLiveData<TSDRingMemory<TSDSignCombination> > de.swp.tsd.trafficsigndetection.TSDSignHistory.getSignCombinations ( ) [inline]
```

Getter for the live data collection of [sign combinations](#). Please do not use the returned reference to directly change the content of the collection but only for registering to as new change listener. Instead use [TSDSignHistory#addSignCombination\(TSDSignCombination, Activity\)](#) to add new combinations.

##### Returns

Collection of sign combinations.

#### 6.43.3.5 isSpeedOk()

```
boolean de.swp.tsd.trafficsigndetection.TSDSignHistory.isSpeedOk (
    Integer speed,
    int threshold ) [inline]
```

This method checks whether the given speed is ok in regards to the information given in the history of recognized signs.

## Parameters

<i>speed</i>	Current speed to check.
<i>threshold</i>	Threshold that the speed can be higher than the limit and still be ok.

## Returns

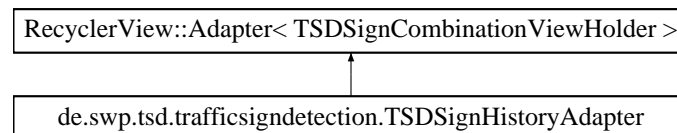
Whether the speed is at most as high as the speed limit plus the threshold.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDSignHistory.java](#)

## 6.44 de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter:



## Classes

- class [DiffCallback](#)

## Public Member Functions

- void [updateTimeDisplay](#) (int from, int to)
- void [onBindViewHolder](#) (@NonNull [TSDSignCombinationViewHolder](#) holder, int position)
- void [onBindViewHolder](#) (@NonNull [TSDSignCombinationViewHolder](#) holder, int position, @NonNull List< Object > payloads)
- int [getItemCount](#) ()
- [TSDSignCombinationViewHolder](#) [onCreateViewHolder](#) (@NonNull ViewGroup parent, int viewType)

## Package Functions

- [TSDSignHistoryAdapter](#) (@NonNull LifecycleOwner lifecycleOwner, @NonNull [TSDSignHistory](#) history, @NonNull LinearLayoutManager layoutManager, @NonNull RecyclerView recyclerView)

### 6.44.1 Detailed Description

This class is an adapter to display all but the latest sign combinations in a given sign history to a RecyclerView. For displaying the single sign combination a [TSDSignCombinationViewHolder](#) is used. The sign itself are drawn transparent to be easily distinguished from the main sign. No speed or warning information are displayed in the history. The display of the history is implemented using a RecyclerView because this type of list view efficiently reuses some views to display the visible part of the list instead of either creating new ones the whole time when the user is scrolling or instead of holding one view for every sign in the collection, which would also be inefficient, since only a portion of them is visible.



## 6.44.2 Constructor & Destructor Documentation

### 6.44.2.1 TSDSignHistoryAdapter()

```
de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.TSDSignHistoryAdapter (
    @NonNull LifecycleOwner lifecycleOwner,
    @NonNull TSDSignHistory history,
    @NonNull LinearLayoutManager layoutManager,
    @NonNull RecyclerView recyclerView ) [inline], [package]
```

This constructor creates a new instance of [TSDSignHistoryAdapter](#) which starts to display the list of given sign combinations in sign card views. For doing this it registers a listener for changes in the history which will listen as long as the given lifecycle owner lives. Every time the history gets updated, the adapter refreshes the internal list and updated the RecyclerView with the new elements, too. As long as the first card is at least partially visible, the view will automatically scroll to the start when a new card is inserted at the front. Otherwise the user would continue to see the old cards when not scrolling to the start of the list himself.

#### Parameters

<i>lifecycleOwner</i>	The lifecycle owner in which this adapter was created so that it does not observe changes in the history for too long.
<i>history</i>	Sign history to display.
<i>layoutManager</i>	The layout manager the adapter items are managed and displayed in.
<i>recyclerView</i>	The recycler view the adapter is used with.

References [de.swp.tsd.trafficsigndetection.TSDRingMemory< T >.getAllSkipLast\(\)](#).

## 6.44.3 Member Function Documentation

### 6.44.3.1 getItemCount()

```
int de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.getItemCount ( ) [inline]
```

Queries for the number of sign combinations which are in the internal sign collection.

#### Returns

Number of sign combinations in the internal collection.

### 6.44.3.2 onBindViewHolder() [1/2]

```
void de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.onBindViewHolder (
    @NonNull TSDSignCombinationViewHolder holder,
    int position ) [inline]
```

This method gets called when the content of a card gets replaced. The given view holder is then told to display the sign at the given position in [TSDSignHistoryAdapter#mSignCombis](#).

## Parameters

<i>holder</i>	View holder to display the selected sign combination in.
<i>position</i>	Position the sign combination is in the collection that should get displayed.

## 6.44.3.3 onBindViewHolder() [2/2]

```
void de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.onBindViewHolder (
    @NonNull TSDSignCombinationViewHolder holder,
    int position,
    @NonNull List< Object > payloads ) [inline]
```

This method gets called when the content of a card gets replaced or updated. The given view holder is then told to display only the changes between the currently displayed sign and the newly selected one to display. This can make the recycler view more efficient. TSDSignHistoryAdapter#mSignCombis.

## Parameters

<i>holder</i>	View holder to display the selected sign combination in.
<i>position</i>	Position the sign combination is in the collection that should get displayed.
<i>payloads</i>	Contains the <a href="#">TSDSignCombination</a> that is currently displayed in the view.

## 6.44.3.4 onCreateViewHolder()

```
TSDSignCombinationViewHolder de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.onCreateViewHolder (
    @NonNull ViewGroup parent,
    int viewType ) [inline]
```

This method gets called when a new [TSDSignCombinationViewHolder](#) gets created and linked to the given view. First the card layout is inflated and then this inflated view is used to create the view holder.

## Parameters

<i>parent</i>	Parent the view should be inflated in, but not attached to its root.
<i>viewType</i>	unused

## Returns

The new view holder which can be used to display sign combinations in.

References [de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.getItemCount\(\)](#).

## 6.44.3.5 updateTimeDisplay()

```
void de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.updateTimeDisplay (
    int from,
    int to ) [inline]
```

This method updates the time display for items in the given inclusive range. This should get called periodically to keep the timespan displayed on the cards up-to-date. Only updating the cards, which are currently visible in the view, can save performance.

## Parameters

<i>from</i>	Inclusive index to update time display in.
<i>to</i>	Inclusive index to update time display in.

References `de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.getItemCount()`.

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

- `App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignHistoryAdapter.java`

## 6.45 de.swp.tsd.trafficsigndetection.TSDSpeedChecker Class Reference

## Public Member Functions

- void `clear` ()
- boolean `isSpeedOk` (Integer speed, int threshold)
- void `addSigns` (@NonNull `TSDSign` [] signs)

## 6.45.1 Detailed Description

This multithreading proof class analyses given signs and classifies given speeds according to the signs in ok or too fast. This is done by checking the incoming signs for opened or ended special speed zones like described in `TSDSign`. When no special zone is open and not speed limit is set, then a base speed limit is used. When a special speed zone is started and no further speed limit is recognized, then the speed limit given by the zone is used. When inside the zone a further speed limit occurs, then this speed limit will get used. If in one combination there are multiple speed limits, then for safety reasons the lowest limit is taken. The multithreading proof is accomplished by using a semaphore to achieve exclusive access for one method call from one thread to the internal variables at a given time.

## 6.45.2 Member Function Documentation

## 6.45.2.1 addSigns()

```
void de.swp.tsd.trafficsigndetection.TSDSpeedChecker.addSigns (
    @NonNull TSDSign [] signs ) [inline]
```

This method refreshes the internal logic for checking the speed by giving it new detected signs.

## Parameters

<i>signs</i>	New signs to add to consideration for speed checking.
--------------	---

## 6.45.2.2 clear()

```
void de.swp.tsd.trafficsigndetection.TSDSpeedChecker.clear ( ) [inline]
```

This method resets all sign related internal saved state. This has the same effect like creating a new instance without having create a new object.

## 6.45.2.3 isSpeedOk()

```
boolean de.swp.tsd.trafficsigndetection.TSDSpeedChecker.isSpeedOk (
    Integer speed,
    int threshold ) [inline]
```

This method checks whether the given speed is under or withing the given threshold of the currently recognized speed limit.

## See also

[TSDSpeedChecker](#)

## Parameters

<i>speed</i>	Current speed.
<i>threshold</i>	Threshold the speed can be higher than the limit.

## Returns

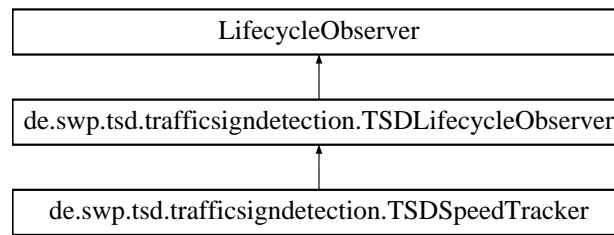
Whether the speed is ok or not.

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

- [App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSpeedChecker.java](#)

## 6.46 de.swp.tsd.trafficsigndetection.TSDSpeedTracker Class Reference

Inheritance diagram for de.swp.tsd.trafficsigndetection.TSDSpeedTracker:



## Classes

- interface [OnSpeedCalculatedListener](#)

## Protected Member Functions

- void [onResume](#) ()
- void [onPause](#) ()

## Package Functions

- [TSDSpeedTracker](#) (@NonNull Lifecycle lifecycle, @NonNull Activity activity, @NonNull [OnSpeedCalculatedListener](#) speedListener)

## Additional Inherited Members

### 6.46.1 Detailed Description

This class defines a location service user which uses the GPS service with the [TSDLocationListener](#) to determine the current speed of the device. It is lifecycle aware using the [TSDLifecycleObserver](#) base class.

See also

[TSDLocationListener](#)

### 6.46.2 Constructor & Destructor Documentation

#### 6.46.2.1 TSDSpeedTracker()

```

de.swp.tsd.trafficsigndetection.TSDSpeedTracker.TSDSpeedTracker (
    @NonNull Lifecycle lifecycle,
    @NonNull Activity activity,
    @NonNull OnSpeedCalculatedListener speedListener ) [inline], [package]
  
```

This constructor creates a new instance of [TSDSpeedTracker](#). This does not enable the speed tracking. To enable the object please see [TSDLifecycleObserver#enable\(\)](#).

## Parameters

<i>lifecycle</i>	Lifecycle of the parent.
<i>activity</i>	Activity the permission request should be handled in.
<i>speedListener</i>	Speed observer being signaled on every speed event.

## 6.46.3 Member Function Documentation

### 6.46.3.1 onPause()

```
void de.swp.tsd.trafficsigndetection.TSDSpeedTracker.onPause ( ) [inline], [protected]
```

This method gets called when the parent lifecycle hits the ON\_PAUSE event and the object is enabled. It stop the location listening and speed tracking.

### 6.46.3.2 onResume()

```
void de.swp.tsd.trafficsigndetection.TSDSpeedTracker.onResume ( ) [inline], [protected]
```

This method gets called when the parent lifecycle hits the ON\_RESUME event and the object is enabled. It initializes and starts the location listening and speed tracking.

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

- App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/[TSDSpeedTracker.java](#)

## Chapter 7

# File Documentation

### 7.1 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/MainActivity.java File Reference

#### Classes

- class [de.swp.tsd.trafficsigndetection.MainActivity](#)

#### Packages

- package [de.swp.tsd.trafficsigndetection](#)

### 7.2 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDAboutFragment.java File Reference ↩↪

#### Classes

- class [de.swp.tsd.trafficsigndetection.TSDAboutFragment](#)

#### Packages

- package [de.swp.tsd.trafficsigndetection](#)

### 7.3 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDAutoFocusMode.java File Reference ↩↪

#### Classes

- enum [de.swp.tsd.trafficsigndetection.TSDAutoFocusMode](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.4 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDBaseCamera.java File Reference

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDBaseCamera](#)
- interface [de.swp.tsd.trafficsigndetection.TSDBaseCamera.OnCameraEventListener](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.5 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraCharacter.java File Reference

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDCameraCharacter](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.6 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraManager.java File Reference

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDCameraManager](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.7 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCameraOutputFormat.java File Reference

### Classes

- enum [de.swp.tsd.trafficsigndetection.TSDCameraOutputFormat](#)



### Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.8 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDCutOffSpeedTracker.java File Reference ↩↪

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker](#)
- interface [de.swp.tsd.trafficsigndetection.TSDCutOffSpeedTracker.OnCutOffSpeedCalculatedListener](#)

### Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.9 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDebugFragment.java File Reference ↩↪

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDDebugFragment](#)

### Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.10 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDetectorFragment.java File Reference ↩↪

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDDetectorFragment](#)

### Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.11 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDisclaimerFragment.java File Reference ↩↪

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDDisclaimerFragment](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.12 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDisclosureFragment.java File Reference

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDDisclosureFragment](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.13 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDrawable.java File Reference

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDDrawable](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.14 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDDrawableDatabase.java File Reference

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDDrawableDatabase](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.15 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDFragment.java File Reference

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDFragment](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.16 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDHelper.java File Reference

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDHelper](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.17 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDInitializationErrorDialog.java File Reference ↩↩

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDInitializationErrorDialog](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.18 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLensFacing.java File Reference ↩↩

## Classes

- enum [de.swp.tsd.trafficsigndetection.TSDLensFacing](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.19 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLicenseFragment.java File Reference ↩↩

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDLicenseFragment](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.20 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLifecycleObserver.java File Reference ↩

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDLifecycleObserver](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.21 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDLocationListener.java File Reference ↩

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDLocationListener](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.22 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDNavigationAdapter.java File Reference ↩

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDNavigationAdapter](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.23 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDOpticalStabilizationMode.java File Reference ↩

### Classes

- enum [de.swp.tsd.trafficsigndetection.TSDOpticalStabilizationMode](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.24 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPeriodicTask.java File Reference ↩↩

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDPeriodicTask](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.25 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPermissionManager.java File Reference ↩↩

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDPermissionManager](#)
- interface [de.swp.tsd.trafficsigndetection.TSDPermissionManager.OnRequestPermissionResultListener](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.26 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDPrivacyFragment.java File Reference ↩↩

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDPrivacyFragment](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.27 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDRingMemory.java File Reference ↩↩

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDRingMemory< T >](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.28 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDRotationSensor.java File Reference↔

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDRotationSensor](#)
- interface [de.swp.tsd.trafficsigndetection.TSDRotationSensor.OnRotationListener](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.29 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSettingsFragment.java File Reference↔

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDSettingsFragment](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.30 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSetupFragment.java File Reference↔

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDSetupFragment](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.31 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDShutterCamera.java File Reference↔

### Classes

- class [de.swp.tsd.trafficsigndetection.TSDShutterCamera](#)

---

Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.32 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDShutterCameraImageListener.java File Reference ↔

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener](#)
- interface [de.swp.tsd.trafficsigndetection.TSDShutterCameraImageListener.FrameEventListener](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.33 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSign.java File Reference

## Classes

- enum [de.swp.tsd.trafficsigndetection.TSDSign](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.34 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignCombination.java File Reference ↔

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDSignCombination](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

## 7.35 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignCombinationViewHolder.java File Reference ↔

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDSignCombinationViewHolder](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

### 7.36 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignHistory.java File Reference

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDSignHistory](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

### 7.37 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSignHistoryAdapter.java File Reference

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter](#)
- class [de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter.DiffCallback](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

### 7.38 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSpeedChecker.java File Reference

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDSpeedChecker](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)

### 7.39 App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/TSDSpeedTracker.java File Reference

## Classes

- class [de.swp.tsd.trafficsigndetection.TSDSpeedTracker](#)
- interface [de.swp.tsd.trafficsigndetection.TSDSpeedTracker.OnSpeedCalculatedListener](#)

## Packages

- package [de.swp.tsd.trafficsigndetection](#)



# Index

[static initializer]  
    de::swp::tsd::trafficsigndetection::MainActivity, 16

A\_8  
    de::swp::tsd::trafficsigndetection::TSDCamera↵  
        OutputFormat, 40

AUTO  
    de::swp::tsd::trafficsigndetection::TSDAutoFocus↵  
        Mode, 27

add  
    de::swp::tsd::trafficsigndetection::TSDRing↵  
        Memory, 85

addAll  
    de::swp::tsd::trafficsigndetection::TSDRing↵  
        Memory, 86

addSignCombination  
    de::swp::tsd::trafficsigndetection::TSDSignHistory,  
        120

addSigns  
    de::swp::tsd::trafficsigndetection::TSDSpeed↵  
        Checker, 125

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    MainActivity.java, 129

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDAboutFragment.java, 129

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDRingMemory.java, 135

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDAutoFocusMode.java, 129

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDRotationSensor.java, 136

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDBaseCamera.java, 130

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSettingsFragment.java, 136

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDCameraCharacter.java, 130

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSetupFragment.java, 136

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDCameraManager.java, 130

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDShutterCamera.java, 136

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDCameraOutputFormat.java, 130

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDShutterCameraImageListener.java, 137

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDCutOffSpeedTracker.java, 131

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSign.java, 137

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDebugFragment.java, 131

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSignCombination.java, 137

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDetectorFragment.java, 131

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSignCombinationViewHolder.java, 137

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDisclaimerFragment.java, 131

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSignHistory.java, 138

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDisclosureFragment.java, 132

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSignHistoryAdapter.java, 138

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDrawable.java, 132

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDrawableDatabase.java, 132

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDFragment.java, 132

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDHelper.java, 133

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDInitializationErrorDialog.java, 133

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDLensFacing.java, 133

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDLicenseFragment.java, 133

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDLifecycleObserver.java, 134

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDLocationListener.java, 134

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDNavigationAdapter.java, 134

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDOpticalStabilizationMode.java, 134

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDPeriodicTask.java, 135

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDPermissionManager.java, 135

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDPrivacyFragment.java, 135

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDRingMemory.java, 135

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDRotationSensor.java, 136

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSettingsFragment.java, 136

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDCameraCharacter.java, 130

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSetupFragment.java, 136

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDCameraManager.java, 130

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDShutterCamera.java, 136

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDCameraOutputFormat.java, 130

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDShutterCameraImageListener.java, 137

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDCutOffSpeedTracker.java, 131

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSign.java, 137

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDebugFragment.java, 131

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSignCombination.java, 137

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDetectorFragment.java, 131

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSignCombinationViewHolder.java, 137

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDisclaimerFragment.java, 131

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSignHistory.java, 138

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDisclosureFragment.java, 132

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSignHistoryAdapter.java, 138

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDSpeedChecker.java, 138

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDrawable.java, 132

App/Android/TSD/app/src/main/java/de/swp/tsd/trafficsigndetection/↵  
    TSDDrawableDatabase.java, 132

- TSDSpeedTracker.java, 138
- areContentsTheSame
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
HistoryAdapter::DiffCallback, 12
- areItemsTheSame
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
HistoryAdapter::DiffCallback, 12
- BACK
  - de::swp::tsd::trafficsigndetection::TSDLensFacing, 67
- betweenResumeAndPause
  - de::swp::tsd::trafficsigndetection::TSDLifecycle↔  
Observer, 70
- CONTINUOUS\_PICTURE
  - de::swp::tsd::trafficsigndetection::TSDAutoFocus↔  
Mode, 27
- CONTINUOUS\_VIDEO
  - de::swp::tsd::trafficsigndetection::TSDAutoFocus↔  
Mode, 27
- clear
  - de::swp::tsd::trafficsigndetection::TSDSpeed↔  
Checker, 126
- createUpdatedBySublist
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, 112
- DEPTH16
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, 40
- DEPTH\_POINT\_CLOUD
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, 40
- de, 9
- de.swp, 9
- de.swp.tsd, 9
- de.swp.tsd.trafficsigndetection, 9
- de.swp.tsd.trafficsigndetection.MainActivity, 15
- de.swp.tsd.trafficsigndetection.TSDAboutFragment, 24
- de.swp.tsd.trafficsigndetection.TSDAutoFocusMode, 25
- de.swp.tsd.trafficsigndetection.TSDBaseCamera, 28
- de.swp.tsd.trafficsigndetection.TSDBaseCamera.On↔  
CameraEventListener, 19
- de.swp.tsd.trafficsigndetection.TSDCameraCharacter, 31
- de.swp.tsd.trafficsigndetection.TSDCameraManager, 36
- de.swp.tsd.trafficsigndetection.TSDCameraOutput↔  
Format, 38
- de.swp.tsd.trafficsigndetection.TSDCutOffSpeed↔  
Tracker, 45
- de.swp.tsd.trafficsigndetection.TSDCutOffSpeed↔  
Tracker.OnCutOffSpeedCalculatedListener, 20
- de.swp.tsd.trafficsigndetection.TSDDebugFragment, 47
- de.swp.tsd.trafficsigndetection.TSDDetectorFragment, 49
- de.swp.tsd.trafficsigndetection.TSDDisclaimerFragment, 52
- de.swp.tsd.trafficsigndetection.TSDDisclosureFragment, 53
- de.swp.tsd.trafficsigndetection.TSDDrawable, 55
- de.swp.tsd.trafficsigndetection.TSDDrawableDatabase, 58
- de.swp.tsd.trafficsigndetection.TSDFragment, 59
- de.swp.tsd.trafficsigndetection.TSDHelper, 61
- de.swp.tsd.trafficsigndetection.TSDInitializationError↔  
Dialog, 65
- de.swp.tsd.trafficsigndetection.TSDLensFacing, 66
- de.swp.tsd.trafficsigndetection.TSDLicenseFragment, 68
- de.swp.tsd.trafficsigndetection.TSDLifecycleObserver, 69
- de.swp.tsd.trafficsigndetection.TSDLocationListener, 72
- de.swp.tsd.trafficsigndetection.TSDNavigationAdapter, 74
- de.swp.tsd.trafficsigndetection.TSDOpticalStabilization↔  
Mode, 77
- de.swp.tsd.trafficsigndetection.TSDPeriodicTask, 79
- de.swp.tsd.trafficsigndetection.TSDPermissionManager, 80
- de.swp.tsd.trafficsigndetection.TSDPermission↔  
Manager.OnRequestPermissionResult↔  
Listener, 21
- de.swp.tsd.trafficsigndetection.TSDPrivacyFragment, 82
- de.swp.tsd.trafficsigndetection.TSDRingMemory< T >, 84
- de.swp.tsd.trafficsigndetection.TSDRotationSensor, 87
- de.swp.tsd.trafficsigndetection.TSDRotationSensor.↔  
OnRotationListener, 23
- de.swp.tsd.trafficsigndetection.TSDSettingsFragment, 89
- de.swp.tsd.trafficsigndetection.TSDSetupFragment, 91
- de.swp.tsd.trafficsigndetection.TSDShutterCamera, 93
- de.swp.tsd.trafficsigndetection.TSDShutterCamera↔  
ImageListener, 96
- de.swp.tsd.trafficsigndetection.TSDShutterCamera↔  
ImageListener.FrameEventListener, 14
- de.swp.tsd.trafficsigndetection.TSDSign, 98
- de.swp.tsd.trafficsigndetection.TSDSignCombination, 110
- de.swp.tsd.trafficsigndetection.TSDSignCombination↔  
ViewHolder, 115
- de.swp.tsd.trafficsigndetection.TSDSignHistory, 119
- de.swp.tsd.trafficsigndetection.TSDSignHistoryAdapter, 122
- de.swp.tsd.trafficsigndetection.TSDSignHistory↔  
Adapter.DiffCallback, 11
- de.swp.tsd.trafficsigndetection.TSDSpeedChecker, 125
- de.swp.tsd.trafficsigndetection.TSDSpeedTracker, 126
- de.swp.tsd.trafficsigndetection.TSDSpeedTracker.On↔  
SpeedCalculatedListener, 23
- de::swp::tsd::trafficsigndetection::MainActivity  
[static initializer], 16
- de::swp::tsd::trafficsigndetection::MainActivity  
getContentLayout, 16
- de::swp::tsd::trafficsigndetection::MainActivity  
onBackPressed, 16
- de::swp::tsd::trafficsigndetection::MainActivity  
onClickMenu, 17

- onCreate, [17](#)
- onDisclaimerAccepted, [17](#)
- onDisclosureAccepted, [18](#)
- onPrivacyPolicyAccepted, [18](#)
- onRequestPermissionsResult, [18](#)
- openDetectorFragment, [19](#)
- de::swp::tsd::trafficsigndetection::TSDAboutFragment
  - onCreateView, [25](#)
  - TSDAboutFragment, [25](#)
- de::swp::tsd::trafficsigndetection::TSDAutoFocusMode
  - AUTO, [27](#)
  - CONTINUOUS\_PICTURE, [27](#)
  - CONTINUOUS\_VIDEO, [27](#)
  - EDOF, [27](#)
  - fromInt, [26](#)
  - MACRO, [27](#)
  - OFF, [27](#)
  - TSDAutoFocusMode, [26](#)
  - value, [28](#)
- de::swp::tsd::trafficsigndetection::TSDBaseCamera
  - getBackgroundHandler, [29](#)
  - getCameraStateCallback, [30](#)
  - getCharacter, [30](#)
  - onPause, [30](#)
  - onResume, [30](#)
  - onTargetsReady, [31](#)
  - TSDBaseCamera, [29](#)
- de::swp::tsd::trafficsigndetection::TSDBaseCamera::↵
  - OnCameraEventListener
  - onClose, [20](#)
  - onOpened, [20](#)
  - onPaused, [20](#)
  - onResume, [20](#)
- de::swp::tsd::trafficsigndetection::TSDCameraCharacter
  - getAvailableApertures, [32](#)
  - getAvailableAutoFocusModes, [32](#)
  - getAvailableOisModes, [33](#)
  - getAvailableOutputFormats, [33](#)
  - getAvailableOutputSizes, [33](#)
  - getCameraCharacteristics, [34](#)
  - getId, [34](#)
  - getLensFacing, [34](#)
  - getManager, [34](#)
  - getMatchingOutputSize, [35](#)
  - getMaximumViewAngle, [35](#)
  - getScore, [35](#)
  - getSensorRotation, [36](#)
  - TSDCameraCharacter, [32](#)
- de::swp::tsd::trafficsigndetection::TSDCameraManager
  - getCameraCharacters, [37](#)
  - openCamera, [38](#)
  - TSDCameraManager, [37](#)
- de::swp::tsd::trafficsigndetection::TSDCameraOutput↵
  - Format
  - A\_8, [40](#)
  - DEPTH16, [40](#)
  - DEPTH\_POINT\_CLOUD, [40](#)
  - FLEX\_RGB\_888, [40](#)
  - FLEX\_RGBA\_8888, [41](#)
  - fromInt, [40](#)
  - GBA\_5551, [41](#)
  - JPEG, [41](#)
  - L\_8, [41](#)
  - LA\_88, [41](#)
  - NV16, [41](#)
  - NV21, [41](#)
  - OPAQUE, [41](#)
  - PRIVATE, [42](#)
  - RAW10, [42](#)
  - RAW12, [42](#)
  - RAW\_PRIVATE, [42](#)
  - RAW\_SENSOR, [42](#)
  - RGB\_332, [42](#)
  - RGB\_565, [42](#)
  - RGB\_888, [42](#)
  - RGBA\_1010102, [43](#)
  - RGBA\_4444, [43](#)
  - RGBA\_8888, [43](#)
  - RGBA\_F16, [43](#)
  - RGBX\_8888, [43](#)
  - TRANSLUCENT, [43](#)
  - TRANSPARENT, [43](#)
  - TSDCameraOutputFormat, [39](#)
  - UNKNOWN, [43](#)
  - value, [44](#)
  - YUV\_420\_888, [44](#)
  - YUV\_422\_888, [44](#)
  - YUV\_444\_888, [44](#)
  - YUY2, [44](#)
  - YV12, [44](#)
- de::swp::tsd::trafficsigndetection::TSDCutOffSpeed↵
  - Tracker
  - onFrameFinished, [46](#)
  - setCameraViewAngle, [46](#)
  - TSDCutOffSpeedTracker, [45](#)
- de::swp::tsd::trafficsigndetection::TSDCutOffSpeed↵
  - Tracker::OnCutOffSpeedCalculatedListener
  - onCutOffSpeedCalculated, [21](#)
- de::swp::tsd::trafficsigndetection::TSDDebugFragment
  - onCreate, [48](#)
  - onCreateView, [48](#)
  - onStart, [49](#)
  - TSDDebugFragment, [47](#)
- de::swp::tsd::trafficsigndetection::TSDDetector↵
  - Fragment
  - onCreate, [50](#)
  - onCreateView, [51](#)
  - onStart, [51](#)
  - TSDDetectorFragment, [50](#)
- de::swp::tsd::trafficsigndetection::TSDDisclaimer↵
  - Fragment
  - onCreateView, [53](#)
  - TSDDisclaimerFragment, [52](#)
- de::swp::tsd::trafficsigndetection::TSDDisclosure↵
  - Fragment
  - onCreateView, [54](#)

- TSDDisclosureFragment, 54
- de::swp::tsd::trafficsigndetection::TSDDrawable
  - equals, 56
  - getDrawable, 57
  - getDrawableId, 57
  - getHeight, 57
  - getWidth, 57
  - TSDDrawable, 55, 56
- de::swp::tsd::trafficsigndetection::TSDDrawable↔
  - Database
  - getDrawable, 58
- de::swp::tsd::trafficsigndetection::TSDFragment
  - displayIconId, 61
  - displayNameId, 61
  - TSDFragment, 59
- de::swp::tsd::trafficsigndetection::TSDHelper
  - dpToPx, 62
  - equalNullable, 63
  - readBooleanFromSharedPreferences, 63
  - readIntegerFromSharedPreferences, 64
  - writeBooleanToSharedPreferences, 64
  - writeIntegerToSharedPreferences, 64
- de::swp::tsd::trafficsigndetection::TSDInitialization↔
  - AlertDialog
  - onCreateDialog, 65
- de::swp::tsd::trafficsigndetection::TSDLensFacing
  - BACK, 67
  - EXTERNAL, 67
  - FRONT, 67
  - fromInt, 67
  - TSDLensFacing, 66
  - value, 67
- de::swp::tsd::trafficsigndetection::TSDLicenseFragment
  - onCreateView, 69
  - TSDLicenseFragment, 68
- de::swp::tsd::trafficsigndetection::TSDLifecycleObserver
  - betweenResumeAndPause, 70
  - disable, 70
  - enable, 71
  - isEnabled, 71
  - onPause, 71
  - onResume, 71
  - onStart, 71
  - TSDLifecycleObserver, 70
- de::swp::tsd::trafficsigndetection::TSDLocationListener
  - onLocationChanged, 73
  - onProviderDisabled, 73
  - onProviderEnabled, 73
  - onStatusChanged, 74
  - TSDLocationListener, 72
- de::swp::tsd::trafficsigndetection::TSDNavigation↔
  - Adapter
  - getCount, 75
  - getItem, 75
  - getItemId, 76
  - getView, 76
  - TSDNavigationAdapter, 75
- de::swp::tsd::trafficsigndetection::TSDOpticalStabilization↔
  - Mode
  - fromInt, 78
  - OFF, 78
  - ON, 78
  - TSDOpticalStabilizationMode, 77
  - value, 78
- de::swp::tsd::trafficsigndetection::TSDPeriodicTask
  - onPause, 80
  - onResume, 80
  - TSDPeriodicTask, 79
- de::swp::tsd::trafficsigndetection::TSDPermission↔
  - Manager
  - getPermissions, 81
  - onRequestPermissionResult, 81
- de::swp::tsd::trafficsigndetection::TSDPermission↔
  - Manager::OnRequestPermissionResult↔
    - Listener
    - getDeclineText, 22
    - getRequestedPermissions, 22
    - onRequestedPermissionsDeclined, 22
    - onRequestedPermissionsGranted, 22
- de::swp::tsd::trafficsigndetection::TSDPrivacyFragment
  - onCreate, 83
  - onCreateView, 83
  - TSDPrivacyFragment, 83
- de::swp::tsd::trafficsigndetection::TSDRingMemory
  - add, 85
  - addAll, 86
  - getAllSkipLast, 86
  - getMaxSize, 87
  - setMaxSize, 87
  - TSDRingMemory, 84
- de::swp::tsd::trafficsigndetection::TSDRotationSensor
  - onPause, 88
  - onResume, 89
  - TSDRotationSensor, 88
- de::swp::tsd::trafficsigndetection::TSDRotationSensor↔
  - ::OnRotationListener
  - onRotation, 23
- de::swp::tsd::trafficsigndetection::TSDSettingsFragment
  - onCreate, 90
  - onCreateView, 90
  - TSDSettingsFragment, 89
- de::swp::tsd::trafficsigndetection::TSDSetupFragment
  - onCreate, 92
  - onCreateView, 92
  - onStart, 93
  - TSDSetupFragment, 91
- de::swp::tsd::trafficsigndetection::TSDShutterCamera
  - disable, 94
  - enable, 94
  - getCameraCharacter, 94
  - getSensorRotation, 95
  - isEnabled, 95
  - isInitiated, 95
  - TSDShutterCamera, 94

- de::swp::tsd::trafficsigndetection::TSDShutterCamera↔
  - ImageListener
  - onImageAvailable, 97
  - roadSignAPIfeedImage, 97
  - setSensorRotation, 98
  - TSDShutterCameraImageListener, 96
- de::swp::tsd::trafficsigndetection::TSDShutterCamera↔
  - ImageListener::FrameEventListener
  - onFrameFinished, 14
- de::swp::tsd::trafficsigndetection::TSDSign
  - endsZone, 100
  - fromId, 101
  - getDrawable, 101
  - isRelevant, 102
  - isStandalone, 102
  - isZoneStart, 102
  - MISC\_BLUE, 103
  - MISC\_BROWN, 103
  - MISC\_GREEN, 103
  - MISC\_RED\_BLUE, 103
  - MISC\_RED\_WHITE, 103
  - MISC\_WHITE, 103
  - MISC\_YELLOW, 103
  - MISC, 102
  - speedLimit, 103
  - TSDSign, 100
  - VZ1000, 104
  - VZ1001\_30, 104
  - VZ1001\_31, 104
  - VZ1001\_32, 104
  - VZ1001\_33, 104
  - VZ1004\_30, 104
  - VZ1004\_31, 104
  - VZ101, 105
  - VZ101\_11, 105
  - VZ103\_10, 105
  - VZ103\_20, 105
  - VZ1040\_30, 105
  - VZ1042\_31, 105
  - VZ1042\_33, 105
  - VZ1053\_35, 105
  - VZ105\_10, 106
  - VZ105\_20, 106
  - VZ120, 106
  - VZ121\_10, 106
  - VZ121\_20, 106
  - VZ123, 106
  - VZ131, 106
  - VZ133\_10, 106
  - VZ136\_10, 107
  - VZ274\_10, 107
  - VZ274\_100, 107
  - VZ274\_110, 107
  - VZ274\_120, 107
  - VZ274\_130, 107
  - VZ274\_20, 107
  - VZ274\_30, 107
  - VZ274\_40, 108
  - VZ274\_50, 108
  - VZ274\_60, 108
  - VZ274\_70, 108
  - VZ274\_80, 108
  - VZ274\_90, 108
  - VZ274x1, 108
  - VZ274x2, 108
  - VZ278, 109
  - VZ282, 109
  - VZ306, 109
  - VZ310, 109
  - VZ311, 109
  - VZ325x1, 109
  - VZ325x2, 109
  - VZ330x1, 109
  - VZ330x2, 110
  - VZ331x1, 110
  - VZ331x2, 110
  - VZ523\_30, 110
- de::swp::tsd::trafficsigndetection::TSDSignCombination
  - createUpdatedBySublist, 112
  - equals, 112
  - getDistanceToImageCenter, 113
  - getFrameId, 113
  - getId, 113
  - getSignDrawables, 113
  - getSigns, 114
  - getSpeed, 114
  - getTimestamp, 114
  - isStandalone, 114
  - isSublistOf, 115
  - setSpeed, 115
  - TSDSignCombination, 111
- de::swp::tsd::trafficsigndetection::TSDSignCombination↔
  - ViewHolder
  - displayAll, 116
  - displayAllChanges, 117
  - displaySigns, 117
  - displaySpeed, 118
  - displayTime, 118
  - setLeftRightMargin, 118
  - setWarningTransparency, 119
  - TSDSignCombinationViewHolder, 116
- de::swp::tsd::trafficsigndetection::TSDSignHistory
  - addSignCombination, 120
  - getLatestFrameId, 120
  - getLatestId, 121
  - getSignCombinations, 121
  - isSpeedOk, 121
  - TSDSignHistory, 120
- de::swp::tsd::trafficsigndetection::TSDSignHistory↔
  - Adapter
  - getItemCount, 123
  - onBindViewHolder, 123, 124
  - onCreateViewHolder, 124
  - TSDSignHistoryAdapter, 123
  - updateTimeDisplay, 124



- de::swp::tsd::trafficsigndetection::TSDSignHistory↔
  - Adapter::DiffCallback
  - areContentsTheSame, 12
  - areItemsTheSame, 12
  - DiffCallback, 11
  - getChangePayload, 13
  - getNewListSize, 13
  - getOldListSize, 13
- de::swp::tsd::trafficsigndetection::TSDSpeedChecker
  - addSigns, 125
  - clear, 126
  - isSpeedOk, 126
- de::swp::tsd::trafficsigndetection::TSDSpeedTracker
  - onPause, 128
  - onResume, 128
  - TSDSpeedTracker, 127
- de::swp::tsd::trafficsigndetection::TSDSpeedTracker::↔
  - OnSpeedCalculatedListener
  - onSpeedCalculated, 24
- DiffCallback
  - de::swp::tsd::trafficsigndetection::TSDSign↔
    - HistoryAdapter::DiffCallback, 11
- disable
  - de::swp::tsd::trafficsigndetection::TSDLifecycle↔
    - Observer, 70
  - de::swp::tsd::trafficsigndetection::TSDShutter↔
    - Camera, 94
- displayAll
  - de::swp::tsd::trafficsigndetection::TSDSign↔
    - CombinationViewHolder, 116
- displayAllChanges
  - de::swp::tsd::trafficsigndetection::TSDSign↔
    - CombinationViewHolder, 117
- displayIconId
  - de::swp::tsd::trafficsigndetection::TSDFragment, 61
- displayNameld
  - de::swp::tsd::trafficsigndetection::TSDFragment, 61
- displaySigns
  - de::swp::tsd::trafficsigndetection::TSDSign↔
    - CombinationViewHolder, 117
- displaySpeed
  - de::swp::tsd::trafficsigndetection::TSDSign↔
    - CombinationViewHolder, 118
- displayTime
  - de::swp::tsd::trafficsigndetection::TSDSign↔
    - CombinationViewHolder, 118
- dpToPx
  - de::swp::tsd::trafficsigndetection::TSDHelper, 62
- EDOF
  - de::swp::tsd::trafficsigndetection::TSDAutoFocus↔
    - Mode, 27
- EXTERNAL
  - de::swp::tsd::trafficsigndetection::TSDLensFacing, 67
- enable
  - de::swp::tsd::trafficsigndetection::TSDLifecycle↔
    - Observer, 71
  - de::swp::tsd::trafficsigndetection::TSDShutter↔
    - Camera, 94
- endsZone
  - de::swp::tsd::trafficsigndetection::TSDSign, 100
- equalNullable
  - de::swp::tsd::trafficsigndetection::TSDHelper, 63
- equals
  - de::swp::tsd::trafficsigndetection::TSDDrawable, 56
  - de::swp::tsd::trafficsigndetection::TSDSign↔
    - Combination, 112
- FLEX\_RGB\_888
  - de::swp::tsd::trafficsigndetection::TSDCamera↔
    - OutputFormat, 40
- FLEX\_RGBA\_8888
  - de::swp::tsd::trafficsigndetection::TSDCamera↔
    - OutputFormat, 41
- FRONT
  - de::swp::tsd::trafficsigndetection::TSDLensFacing, 67
- fromId
  - de::swp::tsd::trafficsigndetection::TSDSign, 101
- fromInt
  - de::swp::tsd::trafficsigndetection::TSDAutoFocus↔
    - Mode, 26
  - de::swp::tsd::trafficsigndetection::TSDCamera↔
    - OutputFormat, 40
  - de::swp::tsd::trafficsigndetection::TSDLensFacing, 67
  - de::swp::tsd::trafficsigndetection::TSDOptical↔
    - StabilizationMode, 78
- GBA\_5551
  - de::swp::tsd::trafficsigndetection::TSDCamera↔
    - OutputFormat, 41
- getAllSkipLast
  - de::swp::tsd::trafficsigndetection::TSDRing↔
    - Memory, 86
- getAvailableApertures
  - de::swp::tsd::trafficsigndetection::TSDCamera↔
    - Character, 32
- getAvailableAutoFocusModes
  - de::swp::tsd::trafficsigndetection::TSDCamera↔
    - Character, 32
- getAvailableOisModes
  - de::swp::tsd::trafficsigndetection::TSDCamera↔
    - Character, 33
- getAvailableOutputFormats
  - de::swp::tsd::trafficsigndetection::TSDCamera↔
    - Character, 33
- getAvailableOutputSizes
  - de::swp::tsd::trafficsigndetection::TSDCamera↔
    - Character, 33
- getBackgroundHandler
  - de::swp::tsd::trafficsigndetection::TSDBase↔
    - Camera, 29

- getCameraCharacter
  - de::swp::tsd::trafficsigndetection::TSDShutter↔  
Camera, [94](#)
- getCameraCharacteristics
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Character, [34](#)
- getCameraCharacters
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Manager, [37](#)
- getCameraStateCallback
  - de::swp::tsd::trafficsigndetection::TSDBase↔  
Camera, [30](#)
- getChangePayload
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
HistoryAdapter::DiffCallback, [13](#)
- getCharacter
  - de::swp::tsd::trafficsigndetection::TSDBase↔  
Camera, [30](#)
- getContentLayout
  - de::swp::tsd::trafficsigndetection::MainActivity, [16](#)
- getCount
  - de::swp::tsd::trafficsigndetection::TSDNavigation↔  
Adapter, [75](#)
- getDeclineText
  - de::swp::tsd::trafficsigndetection::TSDPermission↔  
Manager::OnRequestPermissionResult↔  
Listener, [22](#)
- getDistanceToImageCenter
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, [113](#)
- getDrawable
  - de::swp::tsd::trafficsigndetection::TSDDrawable,  
[57](#)
  - de::swp::tsd::trafficsigndetection::TSDDrawable↔  
Database, [58](#)
  - de::swp::tsd::trafficsigndetection::TSDSign, [101](#)
- getDrawableId
  - de::swp::tsd::trafficsigndetection::TSDDrawable,  
[57](#)
- getFrameId
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, [113](#)
- getHeight
  - de::swp::tsd::trafficsigndetection::TSDDrawable,  
[57](#)
- getId
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Character, [34](#)
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, [113](#)
- getItem
  - de::swp::tsd::trafficsigndetection::TSDNavigation↔  
Adapter, [75](#)
- getItemCount
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
HistoryAdapter, [123](#)
- getItemId
  - de::swp::tsd::trafficsigndetection::TSDNavigation↔  
Adapter, [76](#)
- getLatestFrameId
  - de::swp::tsd::trafficsigndetection::TSDSignHistory,  
[120](#)
- getLatestId
  - de::swp::tsd::trafficsigndetection::TSDSignHistory,  
[121](#)
- getLensFacing
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Character, [34](#)
- getManager
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Character, [34](#)
- getMatchingOutputSize
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Character, [35](#)
- getMaxSize
  - de::swp::tsd::trafficsigndetection::TSDRing↔  
Memory, [87](#)
- getMaximumViewAngle
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Character, [35](#)
- getNewListSize
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
HistoryAdapter::DiffCallback, [13](#)
- getOldListSize
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
HistoryAdapter::DiffCallback, [13](#)
- getPermissions
  - de::swp::tsd::trafficsigndetection::TSDPermission↔  
Manager, [81](#)
- getRequestedPermissions
  - de::swp::tsd::trafficsigndetection::TSDPermission↔  
Manager::OnRequestPermissionResult↔  
Listener, [22](#)
- getScore
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Character, [35](#)
- getSensorRotation
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Character, [36](#)
  - de::swp::tsd::trafficsigndetection::TSDShutter↔  
Camera, [95](#)
- getSignCombinations
  - de::swp::tsd::trafficsigndetection::TSDSignHistory,  
[121](#)
- getSignDrawables
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, [113](#)
- getSigns
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, [114](#)
- getSpeed
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, [114](#)
- getTimestamp
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, [114](#)

- getView
  - de::swp::tsd::trafficsigndetection::TSDNavigation↔ Adapter, 76
- getWidth
  - de::swp::tsd::trafficsigndetection::TSDDrawable, 57
- isEnabled
  - de::swp::tsd::trafficsigndetection::TSDLifecycle↔ Observer, 71
  - de::swp::tsd::trafficsigndetection::TSDShutter↔ Camera, 95
- isInitiated
  - de::swp::tsd::trafficsigndetection::TSDShutter↔ Camera, 95
- isRelevant
  - de::swp::tsd::trafficsigndetection::TSDSign, 102
- isSpeedOk
  - de::swp::tsd::trafficsigndetection::TSDSignHistory, 121
  - de::swp::tsd::trafficsigndetection::TSDSpeed↔ Checker, 126
- isStandalone
  - de::swp::tsd::trafficsigndetection::TSDSign, 102
  - de::swp::tsd::trafficsigndetection::TSDSign↔ Combination, 114
- isSublistOf
  - de::swp::tsd::trafficsigndetection::TSDSign↔ Combination, 115
- isZoneStart
  - de::swp::tsd::trafficsigndetection::TSDSign, 102
- JPEG
  - de::swp::tsd::trafficsigndetection::TSDCamera↔ OutputFormat, 41
- L\_8
  - de::swp::tsd::trafficsigndetection::TSDCamera↔ OutputFormat, 41
- LA\_88
  - de::swp::tsd::trafficsigndetection::TSDCamera↔ OutputFormat, 41
- MACRO
  - de::swp::tsd::trafficsigndetection::TSDAutoFocus↔ Mode, 27
- MISC\_BLUE
  - de::swp::tsd::trafficsigndetection::TSDSign, 103
- MISC\_BROWN
  - de::swp::tsd::trafficsigndetection::TSDSign, 103
- MISC\_GREEN
  - de::swp::tsd::trafficsigndetection::TSDSign, 103
- MISC\_RED\_BLUE
  - de::swp::tsd::trafficsigndetection::TSDSign, 103
- MISC\_RED\_WHITE
  - de::swp::tsd::trafficsigndetection::TSDSign, 103
- MISC\_WHITE
  - de::swp::tsd::trafficsigndetection::TSDSign, 103
- MISC\_YELLOW
  - de::swp::tsd::trafficsigndetection::TSDSign, 103
- MISC
  - de::swp::tsd::trafficsigndetection::TSDSign, 102
- NV16
  - de::swp::tsd::trafficsigndetection::TSDCamera↔ OutputFormat, 41
- NV21
  - de::swp::tsd::trafficsigndetection::TSDCamera↔ OutputFormat, 41
- OFF
  - de::swp::tsd::trafficsigndetection::TSDAutoFocus↔ Mode, 27
  - de::swp::tsd::trafficsigndetection::TSDOptical↔ StabilizationMode, 78
- OPAQUE
  - de::swp::tsd::trafficsigndetection::TSDCamera↔ OutputFormat, 41
- ON
  - de::swp::tsd::trafficsigndetection::TSDOptical↔ StabilizationMode, 78
- onBackPressed
  - de::swp::tsd::trafficsigndetection::MainActivity, 16
- onBindViewHolder
  - de::swp::tsd::trafficsigndetection::TSDSign↔ HistoryAdapter, 123, 124
- onClickMenu
  - de::swp::tsd::trafficsigndetection::MainActivity, 17
- onClose
  - de::swp::tsd::trafficsigndetection::TSDBase↔ Camera::OnCameraEventListener, 20
- onCreate
  - de::swp::tsd::trafficsigndetection::MainActivity, 17
  - de::swp::tsd::trafficsigndetection::TSDDebug↔ Fragment, 48
  - de::swp::tsd::trafficsigndetection::TSDDetector↔ Fragment, 50
  - de::swp::tsd::trafficsigndetection::TSDPrivacy↔ Fragment, 83
  - de::swp::tsd::trafficsigndetection::TSDSettings↔ Fragment, 90
  - de::swp::tsd::trafficsigndetection::TSDSetup↔ Fragment, 92
- onCreateDialog
  - de::swp::tsd::trafficsigndetection::TSDInitialization↔ ErrorDialog, 65
- onCreateView
  - de::swp::tsd::trafficsigndetection::TSDAbout↔ Fragment, 25
  - de::swp::tsd::trafficsigndetection::TSDDebug↔ Fragment, 48
  - de::swp::tsd::trafficsigndetection::TSDDetector↔ Fragment, 51
  - de::swp::tsd::trafficsigndetection::TSDDisclaimer↔ Fragment, 53
  - de::swp::tsd::trafficsigndetection::TSDDisclosure↔ Fragment, 54



- de::swp::tsd::trafficsigndetection::TSDLicense↔  
Fragment, 69
- de::swp::tsd::trafficsigndetection::TSDPrivacy↔  
Fragment, 83
- de::swp::tsd::trafficsigndetection::TSDSettings↔  
Fragment, 90
- de::swp::tsd::trafficsigndetection::TSDSetup↔  
Fragment, 92
- onCreateViewHolder
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
HistoryAdapter, 124
- onCutOffSpeedCalculated
  - de::swp::tsd::trafficsigndetection::TSDCutOff↔  
SpeedTracker::OnCutOffSpeedCalculated↔  
Listener, 21
- onDisclaimerAccepted
  - de::swp::tsd::trafficsigndetection::MainActivity, 17
- onDisclosureAccepted
  - de::swp::tsd::trafficsigndetection::MainActivity, 18
- onFrameFinished
  - de::swp::tsd::trafficsigndetection::TSDCutOff↔  
SpeedTracker, 46
  - de::swp::tsd::trafficsigndetection::TSDShutter↔  
CameraImageListener::FrameEventListener,  
14
- onImageAvailable
  - de::swp::tsd::trafficsigndetection::TSDShutter↔  
CameraImageListener, 97
- onLocationChanged
  - de::swp::tsd::trafficsigndetection::TSDLocation↔  
Listener, 73
- onOpened
  - de::swp::tsd::trafficsigndetection::TSDBase↔  
Camera::OnCameraEventListener, 20
- onPause
  - de::swp::tsd::trafficsigndetection::TSDBase↔  
Camera, 30
  - de::swp::tsd::trafficsigndetection::TSDLifecycle↔  
Observer, 71
  - de::swp::tsd::trafficsigndetection::TSDPeriodic↔  
Task, 80
  - de::swp::tsd::trafficsigndetection::TSDRotation↔  
Sensor, 88
  - de::swp::tsd::trafficsigndetection::TSDSpeed↔  
Tracker, 128
- onPaused
  - de::swp::tsd::trafficsigndetection::TSDBase↔  
Camera::OnCameraEventListener, 20
- onPrivacyPolicyAccepted
  - de::swp::tsd::trafficsigndetection::MainActivity, 18
- onProviderDisabled
  - de::swp::tsd::trafficsigndetection::TSDLocation↔  
Listener, 73
- onProviderEnabled
  - de::swp::tsd::trafficsigndetection::TSDLocation↔  
Listener, 73
- onRequestPermissionsResult
  - de::swp::tsd::trafficsigndetection::TSDPermission↔  
Manager, 81
- onRequestPermissionsResult
  - de::swp::tsd::trafficsigndetection::MainActivity, 18
- onRequestedPermissionsDeclined
  - de::swp::tsd::trafficsigndetection::TSDPermission↔  
Manager::OnRequestPermissionResult↔  
Listener, 22
- onRequestedPermissionsGranted
  - de::swp::tsd::trafficsigndetection::TSDPermission↔  
Manager::OnRequestPermissionResult↔  
Listener, 22
- onResume
  - de::swp::tsd::trafficsigndetection::TSDBase↔  
Camera, 30
  - de::swp::tsd::trafficsigndetection::TSDBase↔  
Camera::OnCameraEventListener, 20
  - de::swp::tsd::trafficsigndetection::TSDLifecycle↔  
Observer, 71
  - de::swp::tsd::trafficsigndetection::TSDPeriodic↔  
Task, 80
  - de::swp::tsd::trafficsigndetection::TSDRotation↔  
Sensor, 89
  - de::swp::tsd::trafficsigndetection::TSDSpeed↔  
Tracker, 128
- onRotation
  - de::swp::tsd::trafficsigndetection::TSDRotation↔  
Sensor::OnRotationListener, 23
- onSpeedCalculated
  - de::swp::tsd::trafficsigndetection::TSDSpeed↔  
Tracker::OnSpeedCalculatedListener, 24
- onStart
  - de::swp::tsd::trafficsigndetection::TSDDebug↔  
Fragment, 49
  - de::swp::tsd::trafficsigndetection::TSDDetector↔  
Fragment, 51
  - de::swp::tsd::trafficsigndetection::TSDLifecycle↔  
Observer, 71
  - de::swp::tsd::trafficsigndetection::TSDSetup↔  
Fragment, 93
- onStatusChanged
  - de::swp::tsd::trafficsigndetection::TSDLocation↔  
Listener, 74
- onTargetsReady
  - de::swp::tsd::trafficsigndetection::TSDBase↔  
Camera, 31
- openCamera
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Manager, 38
- openDetectorFragment
  - de::swp::tsd::trafficsigndetection::MainActivity, 19
- PRIVATE
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, 42
- RAW10
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, 42
- RAW12

- de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [42](#)
- RAW\_PRIVATE
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [42](#)
- RAW\_SENSOR
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [42](#)
- RGB\_332
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [42](#)
- RGB\_565
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [42](#)
- RGB\_888
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [42](#)
- RGBA\_1010102
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [43](#)
- RGBA\_4444
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [43](#)
- RGBA\_8888
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [43](#)
- RGBA\_F16
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [43](#)
- RGBX\_8888
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [43](#)
- readBooleanFromSharedPreferences
  - de::swp::tsd::trafficsigndetection::TSDHelper, [63](#)
- readIntegerFromSharedPreferences
  - de::swp::tsd::trafficsigndetection::TSDHelper, [64](#)
- roadSignAPIfeedImage
  - de::swp::tsd::trafficsigndetection::TSDShutter↔  
CameraImageListener, [97](#)
- setCameraViewAngle
  - de::swp::tsd::trafficsigndetection::TSDCutOff↔  
SpeedTracker, [46](#)
- setLeftRightMargin
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
CombinationViewHolder, [118](#)
- setMaxSize
  - de::swp::tsd::trafficsigndetection::TSDRing↔  
Memory, [87](#)
- setSensorRotation
  - de::swp::tsd::trafficsigndetection::TSDShutter↔  
CameraImageListener, [98](#)
- setSpeed
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, [115](#)
- setWarningTransparency
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
CombinationViewHolder, [119](#)
- speedLimit
  - de::swp::tsd::trafficsigndetection::TSDSign, [103](#)
- TRANSLUCENT
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [43](#)
- TRANSPARENT
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [43](#)
- TSDAboutFragment
  - de::swp::tsd::trafficsigndetection::TSDAbout↔  
Fragment, [25](#)
- TSDAutoFocusMode
  - de::swp::tsd::trafficsigndetection::TSDAutoFocus↔  
Mode, [26](#)
- TSDBaseCamera
  - de::swp::tsd::trafficsigndetection::TSDBase↔  
Camera, [29](#)
- TSDCameraCharacter
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Character, [32](#)
- TSDCameraManager
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
Manager, [37](#)
- TSDCameraOutputFormat
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, [39](#)
- TSDCutOffSpeedTracker
  - de::swp::tsd::trafficsigndetection::TSDCutOff↔  
SpeedTracker, [45](#)
- TSDDebugFragment
  - de::swp::tsd::trafficsigndetection::TSDDebug↔  
Fragment, [47](#)
- TSDDetectorFragment
  - de::swp::tsd::trafficsigndetection::TSDDetector↔  
Fragment, [50](#)
- TSDDisclaimerFragment
  - de::swp::tsd::trafficsigndetection::TSDDisclaimer↔  
Fragment, [52](#)
- TSDDisclosureFragment
  - de::swp::tsd::trafficsigndetection::TSDDisclosure↔  
Fragment, [54](#)
- TSDDrawable
  - de::swp::tsd::trafficsigndetection::TSDDrawable,  
[55](#), [56](#)
- TSDFragment
  - de::swp::tsd::trafficsigndetection::TSDFragment,  
[59](#)
- TSDLensFacing
  - de::swp::tsd::trafficsigndetection::TSDLensFacing,  
[66](#)
- TSDLicenseFragment
  - de::swp::tsd::trafficsigndetection::TSDLicense↔  
Fragment, [68](#)
- TSDLifecycleObserver
  - de::swp::tsd::trafficsigndetection::TSDLifecycle↔  
Observer, [70](#)
- TSDLocationListener
  - de::swp::tsd::trafficsigndetection::TSDLocation↔  
Listener, [72](#)

- TSDNavigationAdapter
  - de::swp::tsd::trafficsigndetection::TSDNavigation↔  
Adapter, 75
- TSDOpticalStabilizationMode
  - de::swp::tsd::trafficsigndetection::TSDOptical↔  
StabilizationMode, 77
- TSDPeriodicTask
  - de::swp::tsd::trafficsigndetection::TSDPeriodic↔  
Task, 79
- TSDPrivacyFragment
  - de::swp::tsd::trafficsigndetection::TSDPrivacy↔  
Fragment, 83
- TSDRingMemory
  - de::swp::tsd::trafficsigndetection::TSDRing↔  
Memory, 84
- TSDRotationSensor
  - de::swp::tsd::trafficsigndetection::TSDRotation↔  
Sensor, 88
- TSDSettingsFragment
  - de::swp::tsd::trafficsigndetection::TSDSettings↔  
Fragment, 89
- TSDSetupFragment
  - de::swp::tsd::trafficsigndetection::TSDSetup↔  
Fragment, 91
- TSDShutterCamera
  - de::swp::tsd::trafficsigndetection::TSDShutter↔  
Camera, 94
- TSDShutterCameraImageListener
  - de::swp::tsd::trafficsigndetection::TSDShutter↔  
CameraImageListener, 96
- TSDSign
  - de::swp::tsd::trafficsigndetection::TSDSign, 100
- TSDSignCombination
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
Combination, 111
- TSDSignCombinationViewHolder
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
CombinationViewHolder, 116
- TSDSignHistory
  - de::swp::tsd::trafficsigndetection::TSDSignHistory,  
120
- TSDSignHistoryAdapter
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
HistoryAdapter, 123
- TSDSpeedTracker
  - de::swp::tsd::trafficsigndetection::TSDSpeed↔  
Tracker, 127
- UNKNOWN
  - de::swp::tsd::trafficsigndetection::TSDCamera↔  
OutputFormat, 43
- updateTimeDisplay
  - de::swp::tsd::trafficsigndetection::TSDSign↔  
HistoryAdapter, 124
- VZ1000
  - de::swp::tsd::trafficsigndetection::TSDSign, 104
- VZ1001\_30
  - de::swp::tsd::trafficsigndetection::TSDSign, 104
- VZ1001\_31
  - de::swp::tsd::trafficsigndetection::TSDSign, 104
- VZ1001\_32
  - de::swp::tsd::trafficsigndetection::TSDSign, 104
- VZ1001\_33
  - de::swp::tsd::trafficsigndetection::TSDSign, 104
- VZ1004\_30
  - de::swp::tsd::trafficsigndetection::TSDSign, 104
- VZ1004\_31
  - de::swp::tsd::trafficsigndetection::TSDSign, 104
- VZ101
  - de::swp::tsd::trafficsigndetection::TSDSign, 105
- VZ101\_11
  - de::swp::tsd::trafficsigndetection::TSDSign, 105
- VZ103\_10
  - de::swp::tsd::trafficsigndetection::TSDSign, 105
- VZ103\_20
  - de::swp::tsd::trafficsigndetection::TSDSign, 105
- VZ1040\_30
  - de::swp::tsd::trafficsigndetection::TSDSign, 105
- VZ1042\_31
  - de::swp::tsd::trafficsigndetection::TSDSign, 105
- VZ1042\_33
  - de::swp::tsd::trafficsigndetection::TSDSign, 105
- VZ1053\_35
  - de::swp::tsd::trafficsigndetection::TSDSign, 105
- VZ105\_10
  - de::swp::tsd::trafficsigndetection::TSDSign, 106
- VZ105\_20
  - de::swp::tsd::trafficsigndetection::TSDSign, 106
- VZ120
  - de::swp::tsd::trafficsigndetection::TSDSign, 106
- VZ121\_10
  - de::swp::tsd::trafficsigndetection::TSDSign, 106
- VZ121\_20
  - de::swp::tsd::trafficsigndetection::TSDSign, 106
- VZ123
  - de::swp::tsd::trafficsigndetection::TSDSign, 106
- VZ131
  - de::swp::tsd::trafficsigndetection::TSDSign, 106
- VZ133\_10
  - de::swp::tsd::trafficsigndetection::TSDSign, 106
- VZ136\_10
  - de::swp::tsd::trafficsigndetection::TSDSign, 107
- VZ274\_10
  - de::swp::tsd::trafficsigndetection::TSDSign, 107
- VZ274\_100
  - de::swp::tsd::trafficsigndetection::TSDSign, 107
- VZ274\_110
  - de::swp::tsd::trafficsigndetection::TSDSign, 107
- VZ274\_120
  - de::swp::tsd::trafficsigndetection::TSDSign, 107
- VZ274\_130
  - de::swp::tsd::trafficsigndetection::TSDSign, 107
- VZ274\_20
  - de::swp::tsd::trafficsigndetection::TSDSign, 107
- VZ274\_30
  - de::swp::tsd::trafficsigndetection::TSDSign, 107

VZ274\_40  
     de::swp::tsd::trafficsigndetection::TSDSign, 108  
 VZ274\_50  
     de::swp::tsd::trafficsigndetection::TSDSign, 108  
 VZ274\_60  
     de::swp::tsd::trafficsigndetection::TSDSign, 108  
 VZ274\_70  
     de::swp::tsd::trafficsigndetection::TSDSign, 108  
 VZ274\_80  
     de::swp::tsd::trafficsigndetection::TSDSign, 108  
 VZ274\_90  
     de::swp::tsd::trafficsigndetection::TSDSign, 108  
 VZ274x1  
     de::swp::tsd::trafficsigndetection::TSDSign, 108  
 VZ274x2  
     de::swp::tsd::trafficsigndetection::TSDSign, 108  
 VZ278  
     de::swp::tsd::trafficsigndetection::TSDSign, 109  
 VZ282  
     de::swp::tsd::trafficsigndetection::TSDSign, 109  
 VZ306  
     de::swp::tsd::trafficsigndetection::TSDSign, 109  
 VZ310  
     de::swp::tsd::trafficsigndetection::TSDSign, 109  
 VZ311  
     de::swp::tsd::trafficsigndetection::TSDSign, 109  
 VZ325x1  
     de::swp::tsd::trafficsigndetection::TSDSign, 109  
 VZ325x2  
     de::swp::tsd::trafficsigndetection::TSDSign, 109  
 VZ330x1  
     de::swp::tsd::trafficsigndetection::TSDSign, 109  
 VZ330x2  
     de::swp::tsd::trafficsigndetection::TSDSign, 110  
 VZ331x1  
     de::swp::tsd::trafficsigndetection::TSDSign, 110  
 VZ331x2  
     de::swp::tsd::trafficsigndetection::TSDSign, 110  
 VZ523\_30  
     de::swp::tsd::trafficsigndetection::TSDSign, 110  
 value  
     de::swp::tsd::trafficsigndetection::TSDAutoFocus↔  
         Mode, 28  
     de::swp::tsd::trafficsigndetection::TSDCamera↔  
         OutputFormat, 44  
     de::swp::tsd::trafficsigndetection::TSDLensFacing,  
         67  
     de::swp::tsd::trafficsigndetection::TSDOptical↔  
         StabilizationMode, 78  
  
 writeBooleanToSharedPreferences  
     de::swp::tsd::trafficsigndetection::TSDHelper, 64  
 writeIntegerToSharedPreferences  
     de::swp::tsd::trafficsigndetection::TSDHelper, 64  
  
 YUV\_420\_888  
     de::swp::tsd::trafficsigndetection::TSDCamera↔  
         OutputFormat, 44  
 YUV\_422\_888  
     de::swp::tsd::trafficsigndetection::TSDCamera↔  
         OutputFormat, 44  
 YUV\_444\_888  
     de::swp::tsd::trafficsigndetection::TSDCamera↔  
         OutputFormat, 44  
 YUY2  
     de::swp::tsd::trafficsigndetection::TSDCamera↔  
         OutputFormat, 44  
 YV12  
     de::swp::tsd::trafficsigndetection::TSDCamera↔  
         OutputFormat, 44