

Reference Manual

Generated by Doxygen 1.8.14

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	9
4.1	File List	9
5	Namespace Documentation	11
5.1	NWH Namespace Reference	11
5.2	NWH.VehiclePhysics Namespace Reference	11
5.3	NWH.WheelController3D Namespace Reference	14
6	Class Documentation	15
6.1	NWH.VehiclePhysics.ABS Class Reference	15
6.1.1	Detailed Description	15
6.1.2	Member Function Documentation	15
6.1.2.1	Update()	15
6.2	NWH.VehiclePhysics.AirBrakeComponent Class Reference	16
6.2.1	Detailed Description	16
6.2.2	Member Function Documentation	16
6.2.2.1	Initialize()	16

6.2.2.2	Update()	16
6.3	NWH.VehiclePhysics.Axle Class Reference	17
6.3.1	Detailed Description	18
6.3.2	Member Enumeration Documentation	18
6.3.2.1	DifferentialType	18
6.3.3	Member Function Documentation	18
6.3.3.1	Initialize()	18
6.3.3.2	TorqueSplit()	19
6.3.3.3	Update()	19
6.3.4	Member Data Documentation	19
6.3.4.1	brakeCoefficient	19
6.3.4.2	differentialStrength	19
6.3.4.3	differentialType	19
6.3.4.4	geometry	20
6.3.4.5	handbrakeCoefficient	20
6.3.4.6	leftWheel	20
6.3.4.7	powerCoefficient	20
6.3.4.8	rightWheel	20
6.3.5	Property Documentation	20
6.3.5.1	Bias	20
6.3.5.2	IsPowered	21
6.3.5.3	NoSlipRPM	21
6.3.5.4	RPM	21
6.3.5.5	SmoothRPM	21
6.3.5.6	WheelSpin	21
6.4	NWH.VehiclePhysics.Backfire Class Reference	21
6.4.1	Detailed Description	22
6.4.2	Member Function Documentation	22
6.4.2.1	Flash()	22
6.4.2.2	Initialize()	22

6.4.2.3	Update()	22
6.4.3	Member Data Documentation	22
6.4.3.1	duration	22
6.4.3.2	renderers	23
6.5	NWH.VehiclePhysics.BackfireComponent Class Reference	23
6.5.1	Detailed Description	23
6.5.2	Member Function Documentation	23
6.5.2.1	Initialize()	23
6.5.2.2	Update()	24
6.6	NWH.VehiclePhysics.BlinkerComponent Class Reference	24
6.6.1	Detailed Description	24
6.6.2	Member Function Documentation	24
6.6.2.1	Initialize()	24
6.6.2.2	Update()	25
6.7	NWH.VehiclePhysics.Brakes Class Reference	25
6.7.1	Member Function Documentation	25
6.7.1.1	Update()	25
6.7.2	Member Data Documentation	26
6.7.2.1	airBrakePressure	26
6.7.2.2	airBrakes	26
6.7.2.3	brakeWhileIdle	26
6.7.2.4	frictionTorque	26
6.7.2.5	maxTorque	26
6.7.2.6	smoothing	26
6.7.3	Property Documentation	27
6.7.3.1	Active	27
6.8	NWH.VehiclePhysics.CameraChanger Class Reference	27
6.8.1	Detailed Description	27
6.8.2	Member Function Documentation	28
6.8.2.1	NextCamera()	28

6.8.3	Member Data Documentation	28
6.8.3.1	cameraTag	28
6.8.3.2	currentCamera	28
6.8.3.3	vehicleCameras	28
6.9	NWH.VehiclePhysics.CameraFollow Class Reference	29
6.9.1	Detailed Description	29
6.9.2	Member Function Documentation	29
6.9.2.1	AngleSigned()	29
6.9.3	Member Data Documentation	30
6.9.3.1	angleFollowStrength	30
6.9.3.2	distance	30
6.9.3.3	height	30
6.9.3.4	smoothing	30
6.9.3.5	target	30
6.9.3.6	targetForwardOffset	31
6.9.3.7	targetUpOffset	31
6.10	NWH.VehiclePhysics.CameraInsideVehicle Class Reference	31
6.10.1	Detailed Description	31
6.11	NWH.VehiclePhysics.CameraMouseDrag Class Reference	31
6.11.1	Detailed Description	32
6.11.2	Member Function Documentation	32
6.11.2.1	ClampAngle()	32
6.11.3	Member Data Documentation	32
6.11.3.1	distance	32
6.11.3.2	distanceSmoothing	33
6.11.3.3	followTargetsRotation	33
6.11.3.4	horizontalMouseSensitivity	33
6.11.3.5	maxDistance	33
6.11.3.6	minDistance	33
6.11.3.7	mouseWheelSensitivity	33

6.11.3.8	smoothing	33
6.11.3.9	target	34
6.11.3.10	verticalMaxAngle	34
6.11.3.11	verticalMinAngle	34
6.11.3.12	verticalMouseSensitivity	34
6.12	NWH.VehiclePhysics.CameraOnboard Class Reference	34
6.12.1	Detailed Description	35
6.12.2	Member Data Documentation	35
6.12.2.1	maxPositionOffsetMagnitude	35
6.12.2.2	positionIntensity	35
6.12.2.3	positionSmoothing	35
6.12.2.4	vehicleController	35
6.13	NWH.VehiclePhysics.CenterOfMass Class Reference	36
6.13.1	Detailed Description	36
6.13.2	Member Data Documentation	36
6.13.2.1	centerOfMassOffset	36
6.13.2.2	showCOM	36
6.14	NWH.VehiclePhysics.CharacterVehicleChanger Class Reference	37
6.14.1	Detailed Description	37
6.14.2	Member Data Documentation	37
6.14.2.1	characterControllerObject	37
6.14.2.2	enterDistance	37
6.14.2.3	enterExitTag	38
6.14.2.4	maxEnterExitVehicleSpeed	38
6.14.2.5	nearVehicle	38
6.15	NWH.VehiclePhysics.DamageHandler.CollisionEvent Class Reference	38
6.15.1	Detailed Description	38
6.15.2	Member Data Documentation	39
6.15.2.1	collision	39
6.15.2.2	decelerationMagnitude	39

6.15.2.3	deformationQueue	39
6.16	NWH.VehiclePhysics.CrashComponent Class Reference	39
6.16.1	Detailed Description	40
6.16.2	Member Function Documentation	40
6.16.2.1	Initialize()	40
6.16.2.2	Play()	40
6.16.2.3	Update()	40
6.17	NWH.VehiclePhysics.DamageHandler Class Reference	41
6.17.1	Detailed Description	42
6.17.2	Member Function Documentation	42
6.17.2.1	Enqueue()	42
6.17.2.2	Initialize()	42
6.17.2.3	MeshDeform()	42
6.17.2.4	Update()	42
6.17.3	Member Data Documentation	42
6.17.3.1	allowedDamage	43
6.17.3.2	decelerationThreshold	43
6.17.3.3	deformationRadius	43
6.17.3.4	deformationRandomness	43
6.17.3.5	deformationStrength	43
6.17.3.6	deformationVerticesPerFrame	43
6.17.3.7	enabled	44
6.17.3.8	ignoreTags	44
6.17.3.9	performanceDegradation	44
6.17.3.10	previousCollisionHash	44
6.17.4	Property Documentation	44
6.17.4.1	Damage	44
6.17.4.2	DamagePercent	45
6.18	NWH.WheelController3D.WheelController.Damper Class Reference	45
6.18.1	Detailed Description	45

6.18.2	Member Data Documentation	45
6.18.2.1	dampingCurve	45
6.18.2.2	force	45
6.18.2.3	maxForce	46
6.18.2.4	unitBumpForce	46
6.18.2.5	unitReboundForce	46
6.19	NWH.VehiclePhysics.DesktopInputManager Class Reference	46
6.19.1	Detailed Description	47
6.19.2	Member Enumeration Documentation	47
6.19.2.1	InputType	47
6.19.3	Member Data Documentation	47
6.19.3.1	inputType	47
6.19.3.2	vehicleChanger	47
6.19.3.3	vehicleController	48
6.20	NWH.VehiclePhysics.Downforce Class Reference	48
6.20.1	Detailed Description	48
6.20.2	Member Data Documentation	48
6.20.2.1	downforcePoints	49
6.20.2.2	maxDownforceSpeed	49
6.21	NWH.VehiclePhysics.Downforce.DownforcePoint Class Reference	49
6.21.1	Detailed Description	49
6.21.2	Member Data Documentation	49
6.21.2.1	maxForce	49
6.21.2.2	position	50
6.22	NWH.VehiclePhysics.DrivingAssists.DrivingAid Class Reference	50
6.22.1	Detailed Description	50
6.22.2	Member Data Documentation	50
6.22.2.1	active	50
6.22.2.2	enabled	51
6.22.2.3	intensity	51

6.23 NWH.VehiclePhysics.DrivingAssists Class Reference	51
6.23.1 Detailed Description	51
6.23.2 Member Function Documentation	52
6.23.2.1 Initialize()	52
6.23.2.2 Update()	52
6.23.3 Member Data Documentation	52
6.23.3.1 abs	52
6.23.3.2 stability	52
6.23.3.3 tcs	52
6.24 NWH.VehiclePhysics.Effects Class Reference	53
6.24.1 Detailed Description	53
6.24.2 Member Function Documentation	53
6.24.2.1 Initialize()	53
6.24.2.2 Update()	53
6.24.3 Member Data Documentation	53
6.24.3.1 exhaustFlash	53
6.24.3.2 exhausts	54
6.24.3.3 lights	54
6.24.3.4 skidmarks	54
6.25 NWH.VehiclePhysics.Engine Class Reference	54
6.25.1 Detailed Description	56
6.25.2 Member Function Documentation	56
6.25.2.1 Initialize()	56
6.25.2.2 Start()	56
6.25.2.3 Stop()	56
6.25.2.4 Toggle()	56
6.25.2.5 Update()	56
6.25.3 Member Data Documentation	57
6.25.3.1 forcedInduction	57
6.25.3.2 maxPower	57

6.25.3.3	maxRPM	57
6.25.3.4	maxRpmChange	57
6.25.3.5	minRPM	57
6.25.3.6	powerCurve	58
6.25.3.7	runOnStartup	58
6.25.3.8	starting	58
6.25.3.9	stopping	58
6.25.3.10	throttleSmoothing	58
6.25.4	Property Documentation	58
6.25.4.1	FuelCutoff	59
6.25.4.2	IsRunning	59
6.25.4.3	Power	59
6.25.4.4	PowerInHP	59
6.25.4.5	RPM	59
6.25.4.6	RpmOverflow	59
6.25.4.7	RPMPercent	60
6.25.4.8	StartingPercent	60
6.25.4.9	StoppingPercent	60
6.25.4.10	TcsPowerReduction	60
6.25.4.11	Torque	60
6.25.4.12	TotalPowerReduction	60
6.26	NWH.VehiclePhysics.EngineIdleSoundComponent Class Reference	61
6.26.1	Detailed Description	61
6.26.2	Member Function Documentation	61
6.26.2.1	Initialize()	61
6.26.2.2	Update()	62
6.26.3	Member Data Documentation	62
6.26.3.1	maxDistortion	62
6.26.3.2	pitchRange	62
6.26.3.3	smoothing	62

6.26.3.4	volumeRange	62
6.27	NWH.VehiclePhysics.EngineStartStopComponent Class Reference	63
6.27.1	Detailed Description	63
6.27.2	Member Function Documentation	63
6.27.2.1	Initialize()	63
6.27.2.2	Update()	63
6.28	NWH.VehiclePhysics.ExhaustSmoke Class Reference	64
6.28.1	Detailed Description	64
6.28.2	Member Function Documentation	64
6.28.2.1	Initialize()	64
6.28.2.2	Update()	65
6.28.3	Member Data Documentation	65
6.28.3.1	baseIntensity	65
6.28.3.2	emit	65
6.28.3.3	intensityRange	65
6.28.3.4	lifeDistance	65
6.28.3.5	particleSystems	65
6.28.3.6	soot	66
6.28.3.7	sootColor	66
6.28.3.8	startSize	66
6.28.3.9	vaporColor	66
6.29	NWH.VehiclePhysics.FlipOver Class Reference	66
6.29.1	Member Function Documentation	67
6.29.1.1	Initialize()	67
6.29.1.2	Update()	67
6.29.2	Member Data Documentation	67
6.29.2.1	allowedAngle	67
6.29.2.2	enabled	67
6.29.2.3	flippedOver	68
6.29.2.4	maxDetectionSpeed	68

6.29.2.5	rotationSpeed	68
6.29.2.6	timeout	68
6.30	NWH.VehiclePhysics.ForcedInduction Class Reference	68
6.30.1	Detailed Description	69
6.30.2	Member Function Documentation	69
6.30.2.1	Initialize()	69
6.30.2.2	Update()	69
6.30.3	Member Data Documentation	69
6.30.3.1	flutterSoundFlag	70
6.30.3.2	maxPowerGainMultiplier	70
6.30.3.3	spoolUpTime	70
6.30.3.4	useForcedInduction	70
6.30.4	Property Documentation	70
6.30.4.1	PowerGainMultiplier	70
6.30.4.2	SpoolPercent	71
6.31	NWH.WheelController3D.WheelController.Friction Class Reference	71
6.31.1	Detailed Description	71
6.31.2	Member Data Documentation	71
6.31.2.1	force	71
6.31.2.2	forceCoefficient	71
6.31.2.3	maxForce	72
6.31.2.4	slip	72
6.31.2.5	slipCoefficient	72
6.31.2.6	speed	72
6.32	NWH.WheelController3D.WheelController.FrictionPreset Class Reference	72
6.32.1	Detailed Description	73
6.32.2	Member Enumeration Documentation	73
6.32.2.1	FrictionPresetEnum	73
6.32.3	Constructor & Destructor Documentation	74
6.32.3.1	FrictionPreset()	74

6.32.4	Member Function Documentation	74
6.32.4.1	GenerateFrictionCurve()	74
6.32.5	Member Data Documentation	74
6.32.5.1	Arcade	74
6.32.5.2	BCDE	74
6.32.5.3	FrictionPresetList	74
6.32.5.4	Generic	75
6.32.5.5	Grass	75
6.32.5.6	Gravel	75
6.32.5.7	Ice	75
6.32.5.8	name	75
6.32.5.9	Sand	75
6.32.5.10	Snow	75
6.32.5.11	TarmacDry	76
6.32.5.12	TarmacWet	76
6.32.5.13	Tracks	76
6.32.6	Property Documentation	76
6.32.6.1	Curve	76
6.33	NWH.VehiclePhysics.Fuel Class Reference	76
6.33.1	Member Function Documentation	77
6.33.1.1	Initialize()	77
6.33.1.2	Update()	77
6.33.2	Member Data Documentation	77
6.33.2.1	amount	77
6.33.2.2	capacity	77
6.33.2.3	efficiency	78
6.33.2.4	useFuel	78
6.33.3	Property Documentation	78
6.33.3.1	ConsumptionKilometersPerLiter	78
6.33.3.2	ConsumptionLitersPer100Kilometers	78

6.33.3.3	ConsumptionLitersPerSecond	78
6.33.3.4	ConsumptionMPG	78
6.33.3.5	FuelPercentage	79
6.33.3.6	HasFuel	79
6.34	NWH.VehiclePhysics.GearChangeComponent Class Reference	79
6.34.1	Detailed Description	79
6.34.2	Member Function Documentation	79
6.34.2.1	Initialize()	80
6.34.2.2	Update()	80
6.35	NWH.VehiclePhysics.Axle.Geometry Class Reference	80
6.35.1	Detailed Description	80
6.35.2	Member Data Documentation	80
6.35.2.1	ackermannPercent	81
6.35.2.2	antiRollBarForce	81
6.35.2.3	camberAtBottom	81
6.35.2.4	camberAtTop	81
6.35.2.5	casterAngle	81
6.35.2.6	isSolid	81
6.35.2.7	steerCoefficient	81
6.35.2.8	toeAngle	82
6.36	NWH.VehiclePhysics.GroundDetection Class Reference	82
6.36.1	Member Function Documentation	82
6.36.1.1	GetCurrentGroundEntity()	83
6.36.1.2	GetCurrentGroundEntityIndex()	83
6.36.1.3	GetCurrentSkidmarkTexture()	83
6.36.2	Member Data Documentation	83
6.36.2.1	dustPrefab	83
6.36.2.2	groundEntities	83
6.36.2.3	smokePrefab	84
6.37	NWH.VehiclePhysics.GroundDetection.GroundEntity Class Reference	84

6.37.1 Detailed Description	84
6.37.2 Member Data Documentation	85
6.37.2.1 dustColor	85
6.37.2.2 dustIntensity	85
6.37.2.3 frictionPresetEnum	85
6.37.2.4 name	85
6.37.2.5 skidmarkMaterial	85
6.37.2.6 skidSoundComponent	86
6.37.2.7 slipBasedSkidIntensity	86
6.37.2.8 slipSensitiveSurfaceSound	86
6.37.2.9 smokeIntensity	86
6.37.2.10 surfaceSoundComponent	86
6.37.2.11 tags	86
6.37.2.12 terrainTextureIndices	87
6.38 NWH.VehiclePhysics.HornComponent Class Reference	87
6.38.1 Detailed Description	87
6.38.2 Member Function Documentation	87
6.38.2.1 Initialize()	87
6.38.2.2 Update()	88
6.39 NWH.VehiclePhysics.InputStates Class Reference	88
6.39.1 Detailed Description	88
6.39.2 Member Function Documentation	89
6.39.2.1 Initialize()	89
6.39.3 Member Data Documentation	89
6.39.3.1 fullBeamLights	89
6.39.3.2 hazardLights	89
6.39.3.3 horn	89
6.39.3.4 leftBlinker	89
6.39.3.5 lowBeamLights	89
6.39.3.6 rightBlinker	90

6.39.3.7	trailerAttachDetach	90
6.39.4	Property Documentation	90
6.39.4.1	Clutch	90
6.39.4.2	Handbrake	90
6.39.4.3	Horizontal	90
6.39.4.4	RawVertical	90
6.39.4.5	ShiftDown	90
6.39.4.6	ShiftUp	91
6.39.4.7	Vertical	91
6.40	NWH.VehiclePhysics.Lights Class Reference	91
6.40.1	Detailed Description	92
6.40.2	Member Function Documentation	92
6.40.2.1	Initialize()	92
6.40.2.2	TurnOffAllLights()	92
6.40.2.3	Update()	92
6.40.3	Member Data Documentation	92
6.40.3.1	enabled	92
6.40.3.2	fullBeams	93
6.40.3.3	headLights	93
6.40.3.4	leftBlinkers	93
6.40.3.5	rearLights	93
6.40.3.6	rightBlinkers	93
6.40.3.7	stopLights	93
6.40.4	Property Documentation	94
6.40.4.1	BlinkerState	94
6.41	NWH.VehiclePhysics.Metrics.Metric Class Reference	94
6.41.1	Member Function Documentation	94
6.41.1.1	Reset()	94
6.41.1.2	Update()	94
6.41.1.3	UpdateDelegate()	95

6.41.2	Member Data Documentation	95
6.41.2.1	value	95
6.42	NWH.VehiclePhysics.Metrics Class Reference	95
6.42.1	Detailed Description	95
6.42.2	Member Function Documentation	96
6.42.2.1	Initialize()	96
6.42.2.2	Update()	96
6.42.3	Member Data Documentation	96
6.42.3.1	averageSpeed	96
6.42.3.2	continousDriftDistance	96
6.42.3.3	continousDriftTime	96
6.42.3.4	odometer	96
6.42.3.5	topSpeed	97
6.42.3.6	totalDriftDistance	97
6.42.3.7	totalDriftTime	97
6.43	NWH.VehiclePhysics.MobileInputManager Class Reference	97
6.43.1	Detailed Description	98
6.43.2	Member Enumeration Documentation	98
6.43.2.1	InputType	98
6.43.3	Member Function Documentation	98
6.43.3.1	BrakeDown()	98
6.43.3.2	BrakeUp()	99
6.43.3.3	ChangeCamera()	99
6.43.3.4	ChangeVehicle()	99
6.43.3.5	EngineStartStop()	99
6.43.3.6	ThrottleDown()	99
6.43.3.7	ThrottleUp()	99
6.43.3.8	ToggleHighBeams()	99
6.43.3.9	ToggleLowBeams()	99
6.43.3.10	TrailerAttachDetach()	100

6.43.3.11	UpdateSteering()	100
6.43.4	Member Data Documentation	100
6.43.4.1	inputType	100
6.43.4.2	steeringWheel	100
6.43.4.3	tiltSensitivity	100
6.43.4.4	vehicleChanger	100
6.43.4.5	vehicleController	101
6.44	NWH.VehiclePhysics.RenderCameraDisable Class Reference	101
6.44.1	Detailed Description	101
6.45	NWH.VehiclePhysics.Rigging Class Reference	101
6.45.1	Member Function Documentation	102
6.45.1.1	Update()	102
6.45.2	Member Data Documentation	102
6.45.2.1	axleBones	102
6.45.2.2	enabled	102
6.45.2.3	solidAxle	102
6.45.2.4	wheelBones	102
6.46	NWH.VehiclePhysics.SkidComponent Class Reference	103
6.46.1	Detailed Description	103
6.46.2	Member Function Documentation	103
6.46.2.1	Initialize()	103
6.46.2.2	Update()	104
6.46.3	Member Data Documentation	104
6.46.3.1	forwardSkidVolume	104
6.46.3.2	sideSkidVolume	104
6.47	NWH.VehiclePhysics.SkidmarkDestroy Class Reference	104
6.47.1	Detailed Description	105
6.47.2	Member Data Documentation	105
6.47.2.1	distanceThreshold	105
6.47.2.2	parentVehicleController	105

6.48 NWH.VehiclePhysics.SkidmarkGenerator Class Reference	105
6.48.1 Detailed Description	106
6.48.2 Member Function Documentation	106
6.48.2.1 CreateNewSnapshot()	106
6.48.2.2 DoubleSubArray()	106
6.48.2.3 Initialize()	106
6.48.2.4 SubArray()	106
6.48.2.5 Update()	106
6.48.3 Member Data Documentation	107
6.48.3.1 groundOffset	107
6.48.3.2 markWidth	107
6.48.3.3 maxMarks	107
6.48.3.4 maxTris	107
6.48.3.5 minSqrDistance	107
6.48.3.6 wheel	107
6.49 NWH.VehiclePhysics.Skidmarks Class Reference	108
6.49.1 Member Function Documentation	108
6.49.1.1 Initialize()	108
6.49.1.2 Update()	108
6.49.2 Member Data Documentation	108
6.49.2.1 enabled	109
6.49.2.2 maxMarksPerSection	109
6.49.2.3 maxSkidmarkAlpha	109
6.49.2.4 minDistance	109
6.49.2.5 persistentSkidmarkDistance	109
6.49.2.6 persistentSkidmarks	109
6.49.2.7 skidmarkStrength	110
6.50 NWH.VehiclePhysics.Sound Class Reference	110
6.50.1 Detailed Description	111
6.50.2 Member Function Documentation	111

6.50.2.1	Disable()	111
6.50.2.2	Enable()	112
6.50.2.3	Initialize()	112
6.50.2.4	SetAudioSourceDefaults()	112
6.50.2.5	SetDefaults()	112
6.50.2.6	Update()	112
6.50.3	Member Data Documentation	113
6.50.3.1	airBrakeComponent	113
6.50.3.2	blinkerComponent	113
6.50.3.3	crashComponent	113
6.50.3.4	crashMixerGroup	113
6.50.3.5	engineIdleComponent	113
6.50.3.6	engineMixerGroup	113
6.50.3.7	engineStartStopComponent	114
6.50.3.8	exhaustPopComponent	114
6.50.3.9	gearChangeComponent	114
6.50.3.10	hornComponent	114
6.50.3.11	insideVehicle	114
6.50.3.12	interiorAttenuation	114
6.50.3.13	lowPassFrequency	115
6.50.3.14	lowPassQ	115
6.50.3.15	masterGroup	115
6.50.3.16	masterVolume	115
6.50.3.17	otherMixerGroup	115
6.50.3.18	skidComponent	115
6.50.3.19	spatialBlend	115
6.50.3.20	surfaceComponent	116
6.50.3.21	surfaceNoiseMixerGroup	116
6.50.3.22	suspensionComponent	116
6.50.3.23	suspensionMixerGroup	116

6.50.3.24 transmissionMixerGroup	116
6.50.3.25 transmissionWhineComponent	116
6.50.3.26 turboFlutterComponent	116
6.50.3.27 turboMixerGroup	117
6.50.3.28 turboWhistleComponent	117
6.51 NWH.VehiclePhysics.GroundDetection.SoundComponent Class Reference	117
6.51.1 Detailed Description	117
6.51.2 Member Data Documentation	117
6.51.2.1 clip	117
6.51.2.2 pitch	118
6.51.2.3 source	118
6.51.2.4 volume	118
6.52 NWH.VehiclePhysics.SoundComponent Class Reference	118
6.52.1 Detailed Description	119
6.52.2 Member Function Documentation	120
6.52.2.1 Disable()	120
6.52.2.2 Enable()	120
6.52.2.3 Initialize()	120
6.52.2.4 RegisterSources()	120
6.52.2.5 SetVolume() [1/2]	120
6.52.2.6 SetVolume() [2/2]	121
6.52.2.7 Update()	121
6.52.3 Member Data Documentation	121
6.52.3.1 audioMixerGroup	121
6.52.3.2 clips	121
6.52.3.3 pitch	121
6.52.3.4 sources	122
6.52.3.5 vc	122
6.52.3.6 volume	122
6.52.4 Property Documentation	122

6.52.4.1	Clip	122
6.52.4.2	Clips	122
6.52.4.3	RandomClip	122
6.52.4.4	Source	123
6.52.4.5	Sources	123
6.53	NWH.WheelController3D.WheelController.Spring Class Reference	123
6.53.1	Detailed Description	123
6.53.2	Member Data Documentation	123
6.53.2.1	bottomedOut	124
6.53.2.2	bottomOutForce	124
6.53.2.3	compressionPercent	124
6.53.2.4	force	124
6.53.2.5	forceCurve	124
6.53.2.6	length	124
6.53.2.7	maxForce	124
6.53.2.8	maxLength	124
6.53.2.9	overExtended	125
6.53.2.10	overflow	125
6.53.2.11	overflowVelocity	125
6.53.2.12	prevLength	125
6.53.2.13	prevOverflow	125
6.53.2.14	targetPoint	125
6.53.2.15	velocity	125
6.54	NWH.VehiclePhysics.Stability Class Reference	126
6.54.1	Detailed Description	126
6.54.2	Member Function Documentation	126
6.54.2.1	Update()	126
6.55	NWH.VehiclePhysics.Steering Class Reference	126
6.55.1	Detailed Description	127
6.55.2	Member Function Documentation	127

6.55.2.1	AdjustGeometry()	127
6.55.2.2	Initialize()	127
6.55.2.3	Steer()	128
6.55.3	Member Data Documentation	128
6.55.3.1	crossoverSpeed	128
6.55.3.2	degreesPerSecondLimit	128
6.55.3.3	highSpeedAngle	128
6.55.3.4	lowSpeedAngle	128
6.55.3.5	steeringWheel	128
6.55.3.6	steeringWheelTurnRatio	129
6.55.3.7	trackedSteerIntensity	129
6.55.4	Property Documentation	129
6.55.4.1	Angle	129
6.56	NWH.VehiclePhysics.SteeringWheel Class Reference	129
6.56.1	Detailed Description	130
6.56.2	Member Function Documentation	130
6.56.2.1	DragEvent()	130
6.56.2.2	GetClampedValue()	130
6.56.2.3	PressEvent()	130
6.56.2.4	ReleaseEvent()	130
6.56.3	Member Data Documentation	131
6.56.3.1	maximumSteeringAngle	131
6.56.3.2	returnToCenterSpeed	131
6.56.3.3	steeringWheelGraphic	131
6.57	NWH.VehiclePhysics.SurfaceComponent Class Reference	131
6.57.1	Detailed Description	132
6.57.2	Member Function Documentation	132
6.57.2.1	Initialize()	132
6.57.2.2	Update()	132
6.58	NWH.VehiclePhysics.SurfaceParticles Class Reference	132

6.58.1 Detailed Description	133
6.58.2 Member Function Documentation	133
6.58.2.1 Initialize()	133
6.58.2.2 Update()	133
6.58.3 Member Data Documentation	133
6.58.3.1 dust	133
6.58.3.2 dustGo	133
6.58.3.3 smoke	133
6.58.3.4 smokeGo	134
6.59 NWH.VehiclePhysics.SuspensionComponent Class Reference	134
6.59.1 Detailed Description	134
6.59.2 Member Function Documentation	134
6.59.2.1 Initialize()	134
6.59.2.2 Update()	135
6.60 NWH.VehiclePhysics.Tracks Class Reference	135
6.60.1 Detailed Description	135
6.60.2 Member Function Documentation	135
6.60.2.1 Initialize()	136
6.60.2.2 Update()	136
6.60.3 Member Data Documentation	136
6.60.3.1 leftSprockets	136
6.60.3.2 leftTrackRenderer	136
6.60.3.3 rightSprockets	136
6.60.3.4 rightTrackRenderer	136
6.60.3.5 textureOffsetSpeedCoefficient	137
6.60.3.6 trackedVehicle	137
6.61 NWH.VehiclePhysics.TractionControl Class Reference	137
6.61.1 Detailed Description	137
6.61.2 Member Function Documentation	138
6.61.2.1 Update()	138

6.61.3	Member Data Documentation	138
6.61.3.1	powerReduction	138
6.61.3.2	prevPowerReduction	138
6.62	NWH.VehiclePhysics.TrailerHandler Class Reference	138
6.62.1	Detailed Description	139
6.62.2	Member Function Documentation	139
6.62.2.1	Initialize()	139
6.62.2.2	Update()	139
6.62.3	Member Data Documentation	140
6.62.3.1	attachDistanceThreshold	140
6.62.3.2	attached	140
6.62.3.3	attachmentPoint	140
6.62.3.4	attachOnPlay	140
6.62.3.5	breakForce	140
6.62.3.6	isTrailer	141
6.62.3.7	maxNoTrailerPowerReduction	141
6.62.3.8	trailerInRange	141
6.62.3.9	trailersTag	141
6.62.3.10	trailerStand	141
6.62.4	Property Documentation	141
6.62.4.1	NoTrailerPowerReduction	142
6.63	NWH.VehiclePhysics.Transmission Class Reference	142
6.63.1	Detailed Description	144
6.63.2	Member Enumeration Documentation	144
6.63.2.1	DifferentialType	144
6.63.2.2	TransmissionType	145
6.63.3	Member Function Documentation	145
6.63.3.1	GetClutchEngagementAtPedalPosition()	145
6.63.3.2	GetGearRatio()	145
6.63.3.3	GetMaxSpeedForGear()	145

6.63.3.4	Initialize()	146
6.63.3.5	ReconstructGearList()	146
6.63.3.6	ReverseTransmitRPM() [1/2]	146
6.63.3.7	ReverseTransmitRPM() [2/2]	146
6.63.3.8	ReverseTransmitTorque()	146
6.63.3.9	ShiftInto()	146
6.63.3.10	TorqueSplit()	146
6.63.3.11	TransmitRPM()	147
6.63.3.12	TransmitTorque()	147
6.63.3.13	Update()	147
6.63.3.14	UpdateClutch()	147
6.63.4	Member Data Documentation	147
6.63.4.1	automaticClutch	147
6.63.4.2	clutchEngagementCurve	148
6.63.4.3	clutchPedalPressedPercent	148
6.63.4.4	differentialType	148
6.63.4.5	dynamicShiftPoint	148
6.63.4.6	gearMultiplier	148
6.63.4.7	lastShiftTime	149
6.63.4.8	postShiftBan	149
6.63.4.9	shiftDuration	149
6.63.4.10	shiftDurationRandomness	149
6.63.4.11	shiftPointRandomness	149
6.63.4.12	targetClutchRPM	149
6.63.4.13	targetShiftDownRPM	150
6.63.4.14	targetShiftUpRPM	150
6.63.4.15	transmissionType	150
6.63.5	Property Documentation	150
6.63.5.1	AddedClutchRPM	150
6.63.5.2	AdjustedShiftDownRpm	150

6.63.5.3	AdjustedShiftUpRpm	151
6.63.5.4	CanShift	151
6.63.5.5	ClutchPercent	151
6.63.5.6	ForwardGearCount	151
6.63.5.7	ForwardGears	151
6.63.5.8	Gear	151
6.63.5.9	GearName	152
6.63.5.10	GearRatio	152
6.63.5.11	Gears	152
6.63.5.12	ReverseGearCount	152
6.63.5.13	ReverseGears	152
6.63.5.14	ReverseRPM	152
6.63.5.15	RPM	153
6.63.5.16	Shifting	153
6.64	NWH.VehiclePhysics.TransmissionWhineComponent Class Reference	153
6.64.1	Member Function Documentation	153
6.64.1.1	Initialize()	154
6.64.1.2	Update()	154
6.64.2	Member Data Documentation	154
6.64.2.1	pitchRange	154
6.65	NWH.VehiclePhysics.TurboFlutterComponent Class Reference	154
6.65.1	Detailed Description	155
6.65.2	Member Function Documentation	155
6.65.2.1	Initialize()	155
6.65.2.2	Update()	155
6.66	NWH.VehiclePhysics.TurboWhistleComponent Class Reference	155
6.66.1	Detailed Description	156
6.66.2	Member Function Documentation	156
6.66.2.1	Initialize()	156
6.66.2.2	Update()	156

6.66.3	Member Data Documentation	156
6.66.3.1	pitchRange	156
6.67	NWH.VehiclePhysics.VehicleChanger Class Reference	157
6.67.1	Detailed Description	157
6.67.2	Member Function Documentation	157
6.67.2.1	ChangeVehicle() [1/2]	158
6.67.2.2	ChangeVehicle() [2/2]	158
6.67.2.3	NearestVehicleFrom()	158
6.67.2.4	NextVehicle()	158
6.67.3	Member Data Documentation	158
6.67.3.1	vehicles	158
6.67.3.2	vehicleTag	159
6.67.4	Property Documentation	159
6.67.4.1	ActiveVehicleController	159
6.67.4.2	CharacterBased	159
6.67.4.3	CurrentVehicle	159
6.67.4.4	DeactivateAll	159
6.68	NWH.VehiclePhysics.VehicleController Class Reference	160
6.68.1	Detailed Description	162
6.68.2	Member Enumeration Documentation	162
6.68.2.1	VehicleCollisionState	162
6.68.3	Member Function Documentation	162
6.68.3.1	AngleSigned()	162
6.68.3.2	DetectWheelSkid()	163
6.68.3.3	DetectWheelSpin()	163
6.68.3.4	GetAverageWheelRPM()	163
6.68.3.5	GetAxles()	163
6.68.3.6	GetCollisionState()	163
6.68.3.7	GetCorrectWheelRpm()	164
6.68.3.8	GetMouseHorizontal()	164

6.68.3.9	GetMouseVertical()	164
6.68.3.10	GetRawAverageWheelRPM()	164
6.68.3.11	LateUpdate()	164
6.68.3.12	Reset()	164
6.68.3.13	ResetCOM()	165
6.68.3.14	SetDefaults()	165
6.68.3.15	Update()	165
6.68.4	Member Data Documentation	165
6.68.4.1	axles	165
6.68.4.2	brakes	165
6.68.4.3	damage	165
6.68.4.4	drivingAids	166
6.68.4.5	effects	166
6.68.4.6	engine	166
6.68.4.7	flipOver	166
6.68.4.8	forwardSlipThreshold	166
6.68.4.9	fuel	166
6.68.4.10	groundDetection	166
6.68.4.11	input	166
6.68.4.12	metrics	167
6.68.4.13	rigging	167
6.68.4.14	sideSlipThreshold	167
6.68.4.15	sound	167
6.68.4.16	speedLimiter	167
6.68.4.17	steering	167
6.68.4.18	tracks	167
6.68.4.19	trailer	167
6.68.4.20	transmission	168
6.68.4.21	vehicleRigidbody	168
6.68.5	Property Documentation	168

6.68.5.1	Acceleration	168
6.68.5.2	Active	168
6.68.5.3	CollisionInfo	168
6.68.5.4	CollisionState	168
6.68.5.5	Direction	169
6.68.5.6	ForwardAcceleration	169
6.68.5.7	ForwardVelocity	169
6.68.5.8	Load	169
6.68.5.9	Speed	169
6.68.5.10	SpeedKPH	169
6.68.5.11	SpeedMPH	170
6.68.5.12	Wheels	170
6.68.5.13	WheelSkid	170
6.68.5.14	WheelSpeed	170
6.68.5.15	WheelSpin	170
6.69	NWH.VehiclePhysics.VehicleControllerInspector Class Reference	171
6.69.1	Detailed Description	171
6.69.2	Member Function Documentation	171
6.69.2.1	OnInspectorGUI()	171
6.70	NWH.VehiclePhysics.VehicleFromScript Class Reference	171
6.70.1	Detailed Description	172
6.70.2	Member Data Documentation	172
6.70.2.1	ignoreLayer	172
6.70.2.2	vehicleMass	172
6.70.2.3	wheelRadius	172
6.70.2.4	wheelTag	172
6.70.2.5	wheelWidth	173
6.71	NWH.VehiclePhysics.Lights.VehicleLight Class Reference	173
6.71.1	Detailed Description	173
6.71.2	Member Function Documentation	173

6.71.2.1	TurnOff()	174
6.71.2.2	TurnOn()	174
6.71.3	Member Data Documentation	174
6.71.3.1	active	174
6.71.3.2	lightMeshRenderers	174
6.71.3.3	lightSources	174
6.71.4	Property Documentation	174
6.71.4.1	On	175
6.72	NWH.VehiclePhysics.Wheel Class Reference	175
6.72.1	Detailed Description	177
6.72.2	Constructor & Destructor Documentation	177
6.72.2.1	Wheel() [1/2]	177
6.72.2.2	Wheel() [2/2]	177
6.72.3	Member Function Documentation	177
6.72.3.1	Activate()	177
6.72.3.2	AddBrakeTorque()	177
6.72.3.3	Initialize()	178
6.72.3.4	Lockup()	178
6.72.3.5	ResetBrakes()	178
6.72.3.6	SetBrakeIntensity()	178
6.72.3.7	Suspend()	178
6.72.3.8	Update()	179
6.72.4	Member Data Documentation	179
6.72.4.1	brakeCoefficient	179
6.72.4.2	wheelController	179
6.72.5	Property Documentation	179
6.72.5.1	Bias	179
6.72.5.2	BrakeTorque	179
6.72.5.3	ControllerGO	179
6.72.5.4	ControllerTransform	180

6.72.5.5	CurrentGroundEntity	180
6.72.5.6	CurrentGroundEntityName	180
6.72.5.7	Damage	180
6.72.5.8	DamageSteerDirection	180
6.72.5.9	ForwardSlip	180
6.72.5.10	ForwardSlipPercent	181
6.72.5.11	HasForwardSlip	181
6.72.5.12	HasSideSlip	181
6.72.5.13	IsGrounded	181
6.72.5.14	MotorTorque	181
6.72.5.15	NoSlipRPM	181
6.72.5.16	Radius	182
6.72.5.17	RPM	182
6.72.5.18	SideSlip	182
6.72.5.19	SideSlipPercent	182
6.72.5.20	SmoothForwardSlip	182
6.72.5.21	SmoothRPM	182
6.72.5.22	SmoothSideSlip	183
6.72.5.23	SpringTravel	183
6.72.5.24	SteerAngle	183
6.72.5.25	VisualTransform	183
6.72.5.26	WheelController	183
6.72.5.27	Width	183
6.73	NWH.WheelController3D.WheelController.Wheel Class Reference	184
6.73.1	Detailed Description	184
6.73.2	Member Function Documentation	185
6.73.2.1	GenerateCamberCurve()	185
6.73.2.2	Initialize()	185
6.73.3	Member Data Documentation	185
6.73.3.1	acceleration	185

6.73.3.2	angularVelocity	185
6.73.3.3	brakeTorque	185
6.73.3.4	camberAngle	185
6.73.3.5	camberCurve	186
6.73.3.6	forward	186
6.73.3.7	freeRollingAngularVelocity	186
6.73.3.8	inertia	186
6.73.3.9	inside	186
6.73.3.10	mass	186
6.73.3.11	motorTorque	186
6.73.3.12	nonRotating	186
6.73.3.13	nonRotatingPostionOffset	187
6.73.3.14	prevGroundPoint	187
6.73.3.15	prevVelocity	187
6.73.3.16	prevWorldPosition	187
6.73.3.17	residualAngularVelocity	187
6.73.3.18	right	187
6.73.3.19	rim	187
6.73.3.20	rimCollider	187
6.73.3.21	rimOffset	188
6.73.3.22	rotationAngle	188
6.73.3.23	rpm	188
6.73.3.24	steerAngle	188
6.73.3.25	tireLoad	188
6.73.3.26	tireRadius	188
6.73.3.27	up	188
6.73.3.28	velocity	188
6.73.3.29	visual	189
6.73.3.30	width	189
6.73.3.31	worldPosition	189

6.73.3.32 worldRotation	189
6.74 NWH.WheelController3D.WheelController Class Reference	189
6.74.1 Detailed Description	192
6.74.2 Member Enumeration Documentation	192
6.74.2.1 Side	192
6.74.3 Member Function Documentation	193
6.74.3.1 DetermineSide()	193
6.74.3.2 FixedUpdate()	193
6.74.3.3 GenerateRimColliderMesh()	193
6.74.3.4 GetFrictionPreset()	193
6.74.3.5 GetGroundHit()	193
6.74.3.6 GetWorldPose()	194
6.74.3.7 Initialize()	194
6.74.3.8 SetActiveFrictionPreset() [1/2]	194
6.74.3.9 SetActiveFrictionPreset() [2/2]	194
6.74.3.10 SetCamber() [1/3]	194
6.74.3.11 SetCamber() [2/3]	195
6.74.3.12 SetCamber() [3/3]	195
6.74.3.13 Start()	195
6.74.4 Member Data Documentation	195
6.74.4.1 activeFrictionPreset	195
6.74.4.2 activeFrictionPresetEnum	196
6.74.4.3 applyForceToOthers	196
6.74.4.4 damper	196
6.74.4.5 debug	196
6.74.4.6 fFriction	196
6.74.4.7 maxPutDownForce	196
6.74.4.8 parent	196
6.74.4.9 sFriction	197
6.74.4.10 singleRay	197

6.74.4.11 singleWheelHit	197
6.74.4.12 spring	197
6.74.4.13 trackedVehicle	197
6.74.4.14 useRimCollider	197
6.74.4.15 wheel	197
6.74.4.16 wheelHit	198
6.74.5 Property Documentation	198
6.74.5.1 angularVelocity	198
6.74.5.2 brakeTorque	198
6.74.5.3 camber	198
6.74.5.4 damperCurve	198
6.74.5.5 damperForce	198
6.74.5.6 damperUnitBumpForce	199
6.74.5.7 damperUnitReboundForce	199
6.74.5.8 forwardFriction	199
6.74.5.9 ForwardScanResolution	199
6.74.5.10 isGrounded	199
6.74.5.11 mass	199
6.74.5.12 MaxPutDownForce	200
6.74.5.13 motorTorque	200
6.74.5.14 NonRotating	200
6.74.5.15 Parent	200
6.74.5.16 pointVelocity	200
6.74.5.17 radius	200
6.74.5.18 rimOffset	201
6.74.5.19 rpm	201
6.74.5.20 ScanIgnoreLayers	201
6.74.5.21 sideFriction	201
6.74.5.22 SideToSideScanResolution	201
6.74.5.23 speed	201

6.74.5.24	springBottomedOut	202
6.74.5.25	springCompression	202
6.74.5.26	springCurve	202
6.74.5.27	springLength	202
6.74.5.28	springMaximumForce	202
6.74.5.29	springOverExtended	202
6.74.5.30	springTravel	203
6.74.5.31	springTravelPoint	203
6.74.5.32	springVelocity	203
6.74.5.33	steerAngle	203
6.74.5.34	suspensionForce	203
6.74.5.35	tireRadius	203
6.74.5.36	tireWidth	204
6.74.5.37	VehicleSide	204
6.74.5.38	Visual	204
6.75	NWH.WheelController3D.WheelControllerInspector Class Reference	204
6.75.1	Member Function Documentation	205
6.75.1.1	OnInspectorGUI()	205
6.75.2	Member Data Documentation	205
6.75.2.1	forwardFrictionCurve	205
6.75.2.2	forwardParams	205
6.75.2.3	sideFrictionCurve	205
6.75.2.4	sideParams	205
6.76	NWH.WheelController3D.WheelController.WheelHit Class Reference	205
6.76.1	Detailed Description	206
6.76.2	Constructor & Destructor Documentation	206
6.76.2.1	WheelHit()	206
6.76.3	Member Function Documentation	207
6.76.3.1	Copy()	207
6.76.4	Member Data Documentation	207

6.76.4.1	angleForward	207
6.76.4.2	curvatureOffset	207
6.76.4.3	distanceFromTire	207
6.76.4.4	force	207
6.76.4.5	forwardDir	207
6.76.4.6	forwardSlip	208
6.76.4.7	groundPoint	208
6.76.4.8	offset	208
6.76.4.9	raycastHit	208
6.76.4.10	sidewaysDir	208
6.76.4.11	sidewaysSlip	208
6.76.4.12	valid	208
6.76.4.13	weight	209
6.76.5	Property Documentation	209
6.76.5.1	collider	209
6.76.5.2	normal	209
6.76.5.3	point	209
7	File Documentation	211
7.1	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraChanger.cs File Reference	211
7.2	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraFollow.cs File Reference	211
7.3	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraInsideVehicle.cs File Reference	211
7.4	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraMouseDrag.cs File Reference	212
7.5	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraOnboard.cs File Reference	212
7.6	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Axle.cs File Reference	212
7.7	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Brakes.cs File Reference	213
7.8	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DamageHandler.cs File Reference	213

7.9	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/ABS.cs File Reference	213
7.10	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/DrivingAssists.cs File Reference	213
7.11	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/Stability.cs File Reference	214
7.12	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/TractionControl.cs File Reference	214
7.13	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Editor/VehicleControllerInspector.cs File Reference	214
7.14	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/Backfire.cs File Reference	215
7.15	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/Effects.cs File Reference	215
7.16	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ExhaustSmoke.cs File Reference	215
7.17	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/Lights.cs File Reference	215
7.18	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/SkidmarkDestroy.cs File Reference	216
7.19	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/SkidmarkGenerator.cs File Reference	216
7.20	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/Skidmarks.cs File Reference	216
7.21	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/SurfaceParticles.cs File Reference	217
7.22	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Engine.cs File Reference	217
7.23	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/FlipOver.cs File Reference	217
7.24	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ForcedInduction.cs File Reference	217
7.25	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Fuel.cs File Reference	218
7.26	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/GroundDetection.cs File Reference	218
7.27	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/DesktopInputManager.cs File Reference	218
7.28	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/InputStates.cs File Reference	219
7.29	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/MobileInputManager.cs File Reference	219

7.30	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/SteeringWheel.cs File Reference	219
7.31	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Metrics.cs File Reference	219
7.32	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/CenterOfMass.cs File Reference	220
7.33	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/CharacterVehicle↔Changer.cs File Reference	220
7.34	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/Downforce.cs File Reference	220
7.35	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/RenderCamera↔Disable.cs File Reference	221
7.36	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/UnitConverter.cs File Reference	221
7.37	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/VehicleChanger.cs File Reference	221
7.38	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/VehicleFrom↔Script.cs File Reference	221
7.39	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Rigging.cs File Reference	222
7.40	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/AirBrake↔Component.cs File Reference	222
7.41	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/Backfire↔Component.cs File Reference	222
7.42	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/BlinkerComponent.cs File Reference	223
7.43	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/CrashComponent.cs File Reference	223
7.44	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/EngineIdle↔Component.cs File Reference	223
7.45	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/EngineStartStop↔Component.cs File Reference	223
7.46	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/GearChange↔Component.cs File Reference	224
7.47	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/HornComponent.cs File Reference	224
7.48	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/SkidComponent.cs File Reference	224
7.49	E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/Sound.cs File Reference	225

7.50 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/SoundComponent.cs File Reference	225
7.51 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/Surface↔Component.cs File Reference	225
7.52 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/Suspension↔Component.cs File Reference	225
7.53 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/Transmission↔WhineComponent.cs File Reference	226
7.54 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/TurboFlutter↔Component.cs File Reference	226
7.55 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/TurboWhistle↔Component.cs File Reference	226
7.56 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Steering.cs File Reference	227
7.57 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Tracks.cs File Reference	227
7.58 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/TrailerHandler.cs File Reference	227
7.59 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Transmission.cs File Reference	227
7.60 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/VehicleController.cs File Reference	228
7.61 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Wheel.cs File Reference	228
7.62 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/Editor/WheelController↔Inspector.cs File Reference	228
7.63 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/WheelController.API.cs File Reference	229
7.64 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/WheelController.cs File Reference	229
7.65 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/WheelController.Debug.cs File Reference	229
7.66 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/WheelController.Friction↔Preset.cs File Reference	230

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

NWH	11
NWH.VehiclePhysics	11
NWH.WheelController3D	14

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

NWH.VehiclePhysics.Axle	17
NWH.VehiclePhysics.Backfire	21
NWH.VehiclePhysics.Brakes	25
NWH.VehiclePhysics.DamageHandler.CollisionEvent	38
NWH.VehiclePhysics.DamageHandler	41
NWH.WheelController3D.WheelController.Damper	45
NWH.VehiclePhysics.Downforce.DownforcePoint	49
NWH.VehiclePhysics.DrivingAssists.DrivingAid	50
NWH.VehiclePhysics.ABS	15
NWH.VehiclePhysics.Stability	126
NWH.VehiclePhysics.TractionControl	137
NWH.VehiclePhysics.DrivingAssists	51
Editor	
NWH.VehiclePhysics.VehicleControllerInspector	171
NWH.WheelController3D.WheelControllerInspector	204
NWH.VehiclePhysics.Effects	53
NWH.VehiclePhysics.Engine	54
NWH.VehiclePhysics.ExhaustSmoke	64
NWH.VehiclePhysics.FlipOver	66
NWH.VehiclePhysics.ForcedInduction	68
NWH.WheelController3D.WheelController.Friction	71
NWH.WheelController3D.WheelController.FrictionPreset	72
NWH.VehiclePhysics.Fuel	76
NWH.VehiclePhysics.Axle.Geometry	80
NWH.VehiclePhysics.GroundDetection.GroundEntity	84
NWH.VehiclePhysics.InputStates	88
NWH.VehiclePhysics.Lights	91
NWH.VehiclePhysics.Metrics.Metric	94
NWH.VehiclePhysics.Metrics	95
MonoBehaviour	
NWH.VehiclePhysics.CameraChanger	27
NWH.VehiclePhysics.CameraFollow	29
NWH.VehiclePhysics.CameraInsideVehicle	31
NWH.VehiclePhysics.CameraMouseDrag	31
NWH.VehiclePhysics.CameraOnboard	34

NWH.VehiclePhysics.CenterOfMass	36
NWH.VehiclePhysics.CharacterVehicleChanger	37
NWH.VehiclePhysics.DesktopInputManager	46
NWH.VehiclePhysics.Downforce	48
NWH.VehiclePhysics.GroundDetection	82
NWH.VehiclePhysics.MobileInputManager	97
NWH.VehiclePhysics.RenderCameraDisable	101
NWH.VehiclePhysics.SkidmarkDestroy	104
NWH.VehiclePhysics.SteeringWheel	129
NWH.VehiclePhysics.VehicleChanger	157
NWH.VehiclePhysics.VehicleController	160
NWH.VehiclePhysics.VehicleFromScript	171
NWH.WheelController3D.WheelController	189
NWH.WheelController3D.WheelController	189
NWH.WheelController3D.WheelController	189
NWH.VehiclePhysics.Rigging	101
NWH.VehiclePhysics.SkidmarkGenerator	105
NWH.VehiclePhysics.Skidmarks	108
NWH.VehiclePhysics.Sound	110
NWH.VehiclePhysics.GroundDetection.SoundComponent	117
NWH.VehiclePhysics.SoundComponent	118
NWH.VehiclePhysics.AirBrakeComponent	16
NWH.VehiclePhysics.BackfireComponent	23
NWH.VehiclePhysics.BlinkerComponent	24
NWH.VehiclePhysics.CrashComponent	39
NWH.VehiclePhysics.EngineIdleSoundComponent	61
NWH.VehiclePhysics.EngineStartStopComponent	63
NWH.VehiclePhysics.GearChangeComponent	79
NWH.VehiclePhysics.HornComponent	87
NWH.VehiclePhysics.SkidComponent	103
NWH.VehiclePhysics.SurfaceComponent	131
NWH.VehiclePhysics.SuspensionComponent	134
NWH.VehiclePhysics.TransmissionWhineComponent	153
NWH.VehiclePhysics.TurboFlutterComponent	154
NWH.VehiclePhysics.TurboWhistleComponent	155
NWH.WheelController3D.WheelController.Spring	123
NWH.VehiclePhysics.Steering	126
NWH.VehiclePhysics.SurfaceParticles	132
NWH.VehiclePhysics.Tracks	135
NWH.VehiclePhysics.TrailerHandler	138
NWH.VehiclePhysics.Transmission	142
NWH.VehiclePhysics.Lights.VehicleLight	173
NWH.VehiclePhysics.Wheel	175
NWH.WheelController3D.WheelController.Wheel	184
NWH.WheelController3D.WheelController.WheelHit	205

Chapter 3

Class Index

3.1 Class List

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

NWH.VehiclePhysics.ABS	
Traction control class.	15
NWH.VehiclePhysics.AirBrakeComponent	
Hiss produced by air brakes releasing air. Accepts multiple clips of which one will be chosen at random each time this effect is played.	16
NWH.VehiclePhysics.Axle	
Represents a single axle of a vehicle.	17
NWH.VehiclePhysics.Backfire	
Controls exhaust flames / flashes.	21
NWH.VehiclePhysics.BackfireComponent	
Exhaust popping on deceleration / rev limiter.	23
NWH.VehiclePhysics.BlinkerComponent	
Click-clack of the working blinker. Accepts two clips, first is for the blinker turning on and the second is for blinker turning off.	24
NWH.VehiclePhysics.Brakes	25
NWH.VehiclePhysics.CameraChanger	
Switches between the camera objects that are children to this object and contain camera tag, in order they appear in the hierarchy or in order they are added to the vehicle cameras list.	27
NWH.VehiclePhysics.CameraFollow	
Camera that follows behind the vehicle.	29
NWH.VehiclePhysics.CameraInsideVehicle	
Empty component that should be attached to the cameras that are inside the vehicle if interior sound change is to be used.	31
NWH.VehiclePhysics.CameraMouseDrag	
Camera that can be dragged with the mouse.	31
NWH.VehiclePhysics.CameraOnboard	
Camera for on or in-vehicle use with option of head movement according to the G-forces.	34
NWH.VehiclePhysics.CenterOfMass	
Used for adjusting center of mass of any rigidbody object.	36
NWH.VehiclePhysics.CharacterVehicleChanger	
Allows character to enter or exit vehicle. Can be used with any first or 3rd person object.	37
NWH.VehiclePhysics.DamageHandler.CollisionEvent	
Contains data on the collision that has last happened.	38
NWH.VehiclePhysics.CrashComponent	
Sound of vehicle crashing into an object. Supports multiple audio clips of which one will be chosen at random each time this effect is played.	39

NWH.VehiclePhysics.DamageHandler	
Handles all damage related calculations and mesh deformations. Collision sounds are handled by CrashComponent class.	41
NWH.WheelController3D.WheelController.Damper	
Suspension part.	45
NWH.VehiclePhysics.DesktopInputManager	
Class for handling desktop user input via mouse and keyboard. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.	46
NWH.VehiclePhysics.Downforce	
Loose approximation on downforce acting on a vehicle. Downforce in this case is only dependent on speed, but its amount and speed at which it is achieved can be adjusted. Can be adjusted at runtime.	48
NWH.VehiclePhysics.Downforce.DownforcePoint	
Single point at which downforce will be applied.	49
NWH.VehiclePhysics.DrivingAssists.DrivingAid	
Base class for driving aids.	50
NWH.VehiclePhysics.DrivingAssists	
Systems to help driver with vehicle control.	51
NWH.VehiclePhysics.Effects	
Main class for handling visual effects such as skidmarks, lights and exhausts.	53
NWH.VehiclePhysics.Engine	
Everything related to a vehicle's engine and its systems.	54
NWH.VehiclePhysics.EngineIdleSoundComponent	
Sound of an engine idling.	61
NWH.VehiclePhysics.EngineStartStopComponent	
Sound of an engine starting / stopping. First audio clip is for engine starting, and second one is for engine stopping.	63
NWH.VehiclePhysics.ExhaustSmoke	
Controls particle emitters that represent exhausts.	64
NWH.VehiclePhysics.FlipOver	
.	66
NWH.VehiclePhysics.ForcedInduction	
Supercharger, turbocharger, etc. Can also be used on vehicles with no forced induction for sound effects such as intake noise or engine fan noise.	68
NWH.WheelController3D.WheelController.Friction	
All info related to longitudinal force calculation.	71
NWH.WheelController3D.WheelController.FrictionPreset	
Container class for holding wheel friction presets.	72
NWH.VehiclePhysics.Fuel	
.	76
NWH.VehiclePhysics.GearChangeComponent	
Shifter sound played when changing gears. Supports multiple audio clips of which one is chosen at random each time this effect is played.	79
NWH.VehiclePhysics.Axle.Geometry	
Class holding all geometry related data for axle and its wheels.	80
NWH.VehiclePhysics.GroundDetection	
.	82
NWH.VehiclePhysics.GroundDetection.GroundEntity	
A class representing a single ground surface type.	84
NWH.VehiclePhysics.HornComponent	
Vehicle horn sound.	87
NWH.VehiclePhysics.InputStates	
Class for storing input states of the vehicle.	88
NWH.VehiclePhysics.Lights	
Class for controlling all of the vehicle lights.	91
NWH.VehiclePhysics.Metrics.Metric	
.	94
NWH.VehiclePhysics.Metrics	
Class for holding metrics such as odometer, top speed and drift time. All the speed values are in m/s. If you need the value in km/h or mph use UnitConverter functions.	95

NWH.VehiclePhysics.MobileInputManager	
Class for handling mobile user input via touch screen and sensors. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.	97
NWH.VehiclePhysics.RenderCameraDisable	
Attach this to any cameras rendering to mirrors so that when vehicle is inactive mirror will not update.	101
NWH.VehiclePhysics.Rigging	101
NWH.VehiclePhysics.SkidComponent	
Sound produced by tire skidding over surface.	103
NWH.VehiclePhysics.SkidmarkDestroy	
Destroys skidmark object when distance to the vehicle becomes greater than distance threshold.	104
NWH.VehiclePhysics.SkidmarkGenerator	
Generates skidmark meshes.	105
NWH.VehiclePhysics.Skidmarks	108
NWH.VehiclePhysics.Sound	
Main class that manages all the sound aspects of the vehicle.	110
NWH.VehiclePhysics.GroundDetection.SoundComponent	
Base class for surface sounds.	117
NWH.VehiclePhysics.SoundComponent	
Base abstract class from which all vehicle sound components inherit.	118
NWH.WheelController3D.WheelController.Spring	
Suspension part.	123
NWH.VehiclePhysics.Stability	
Traction control class.	126
NWH.VehiclePhysics.Steering	
Everything related to steering and axle's geometry.	126
NWH.VehiclePhysics.SteeringWheel	
Script for controlling the GUI steering wheel for mobile controls. Credits go to yasirkula from Unity Forums for original code.	129
NWH.VehiclePhysics.SurfaceComponent	
Sounds produced by tire rolling over the surface.	131
NWH.VehiclePhysics.SurfaceParticles	
Skid smoke and surface dust generated by wheel slipping / rolling over the surface.	132
NWH.VehiclePhysics.SuspensionComponent	
Sound of wheel hitting the surface or obstacle.	134
NWH.VehiclePhysics.Tracks	
Class for handling tracked vehicles. If enabled all the wheels should be the same size and placed on either left or right side of the vehicle. Center wheels are not supported in this mode.	135
NWH.VehiclePhysics.TractionControl	
Traction control class.	137
NWH.VehiclePhysics.TrailerHandler	
Everything related to a trailer.	138
NWH.VehiclePhysics.Transmission	
Handles gear changing and also torque and RPM transmission in both directions.	142
NWH.VehiclePhysics.TransmissionWhineComponent	153
NWH.VehiclePhysics.TurboFlutterComponent	
Sound of wastegate releasing air on turbocharged vehicles.	154
NWH.VehiclePhysics.TurboWhistleComponent	
Sound of turbocharger or supercharger.	155
NWH.VehiclePhysics.VehicleChanger	
Used for changing vehicles. Also activates and deactivates vehicle cameras if default VehicleCamera system is used.	157
NWH.VehiclePhysics.VehicleController	
Main class controlling all the other parts of the vehicle.	160
NWH.VehiclePhysics.VehicleControllerInspector	
Shows different settings for trailer depending if vehicle is a trailer or a towing vehicle.	171

NWH.VehiclePhysics.VehicleFromScript	
Example script for setting up a vehicle at runtime - modify as/if needed. Sets up a vehicle from script at runtime. Model is required to already have body colliders set up, as well as wheels tagged with correct tag. Model also needs to have correct rotation (Z-forward, Y-up, X-right). Works only on vehicles with two wheels per axle. Number of axles is not limited.	171
NWH.VehiclePhysics.Lights.VehicleLight	
Single vehicle light.	173
NWH.VehiclePhysics.Wheel	
Contains everything related to wheels. To access WC3D's properties directly use Wheel↔Controller getter. Most used variables are wrapped in getters and setter to enable eventual (but unlikely) future compatibility with default wheel collider. Wheel class is not equal to Wheel↔Controller class. To access WC3D (WheelController) use WheelController getter/setter.	175
NWH.WheelController3D.WheelController.Wheel	
Contains everything wheel related, including rim and tire.	184
NWH.WheelController3D.WheelController	
API for WheelController	189
NWH.WheelController3D.WheelControllerInspector	
NWH.WheelController3D.WheelController.WheelHit	
Contains RaycastHit and extended hit data.	205

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/ CameraChanger.cs	211
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/ CameraFollow.cs	211
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/ CameraInsideVehicle.cs	211
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/ CameraMouseDrag.cs	212
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/ CameraOnboard.cs	212
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Axle.cs	212
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Brakes.cs	213
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ DamageHandler.cs	213
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Engine.cs	217
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ FlipOver.cs	217
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ ForcedInduction.cs	217
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Fuel.cs	218
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ GroundDetection.cs	218
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Metrics.cs	219
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Rigging.cs	222
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Steering.cs	227
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Tracks.cs	227
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ TrailerHandler.cs	227
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Transmission.cs	227
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ VehicleController.cs	228
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ Wheel.cs	228
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/ ABS.cs	213
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/ DrivingAssists.cs	213
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/ Stability.cs	214
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/ TractionControl.cs	214
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Editor/ VehicleControllerInspector.cs	214
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ Backfire.cs	215
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ Effects.cs	215
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ ExhaustSmoke.cs	215
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ Lights.cs	215
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ SkidmarkDestroy.cs	216
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ SkidmarkGenerator.cs	216
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ Skidmarks.cs	216

E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ SurfaceParticles.cs . . .	217
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/ DesktopInputManager.cs	218
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/ InputStates.cs	219
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/ MobileInputManager.cs .	219
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/ SteeringWheel.cs	219
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/ CenterOfMass.cs	220
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/ CharacterVehicleChanger.cs 220	
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/ Downforce.cs	220
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/ RenderCameraDisable.cs	221
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/ UnitConverter.cs	221
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/ VehicleChanger.cs	221
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/ VehicleFromScript.cs . .	221
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ AirBrakeComponent.cs	222
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ BackfireComponent.cs	222
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ BlinkerComponent.cs .	223
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ CrashComponent.cs . .	223
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ EngineIdleComponent.cs 223	
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ EngineStartStopComponent.cs 223	
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ GearChangeComponent.cs 224	
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ HornComponent.cs . .	224
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ SkidComponent.cs . .	224
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ Sound.cs	225
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ SoundComponent.cs .	225
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ SurfaceComponent.cs .	225
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ SuspensionComponent.cs 225	
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ TransmissionWhineComponent.cs 226	
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ TurboFlutterComponent.cs 226	
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/ TurboWhistleComponent.cs 226	
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/ WheelController.API.cs	229
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/ WheelController.cs	229
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/ WheelController.Debug.cs . . .	229
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/ WheelController.FrictionPreset.cs 230	
E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/Editor/ WheelControllerInspector.cs 228	

Chapter 5

Namespace Documentation

5.1 NWH Namespace Reference

Namespaces

- namespace [VehiclePhysics](#)
- namespace [WheelController3D](#)

5.2 NWH.VehiclePhysics Namespace Reference

Classes

- class [ABS](#)
Traction control class.
- class [AirBrakeComponent](#)
Hiss produced by air brakes releasing air. Accepts multiple clips of which one will be chosen at random each time this effect is played.
- class [Axle](#)
Represents a single axle of a vehicle.
- class [Backfire](#)
Controls exhaust flames / flashes.
- class [BackfireComponent](#)
Exhaust popping on deceleration / rev limiter.
- class [BlinkerComponent](#)
Click-clack of the working blinker. Accepts two clips, first is for the blinker turning on and the second is for blinker turning off.
- class [Brakes](#)
- class [CameraChanger](#)
Switches between the camera objects that are children to this object and contain camera tag, in order they appear in the hierarchy or in order they are added to the vehicle cameras list.
- class [CameraFollow](#)
Camera that follows behind the vehicle.
- class [CameraInsideVehicle](#)
Empty component that should be attached to the cameras that are inside the vehicle if interior sound change is to be used.

- class [CameraMouseDrag](#)
Camera that can be dragged with the mouse.
- class [CameraOnboard](#)
Camera for on or in-vehicle use with option of head movement according to the G-forces.
- class [CenterOfMass](#)
Used for adjusting center of mass of any rigidbody object.
- class [CharacterVehicleChanger](#)
Allows character to enter or exit vehicle. Can be used with any first or 3rd person object.
- class [CrashComponent](#)
Sound of vehicle crashing into an object. Supports multiple audio clips of which one will be chosen at random each time this effect is played.
- class [DamageHandler](#)
Handles all damage related calculations and mesh deformations. Collision sounds are handled by [CrashComponent](#) class.
- class [DesktopInputManager](#)
Class for handling desktop user input via mouse and keyboard. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.
- class [Downforce](#)
Loose approximation on downforce acting on a vehicle. [Downforce](#) in this case is only dependent on speed, but its amount and speed at which it is achieved can be adjusted. Can be adjusted at runtime.
- class [DrivingAssists](#)
Systems to help driver with vehicle control.
- class [Effects](#)
Main class for handling visual effects such as skidmarks, lights and exhausts.
- class [Engine](#)
Everything related to a vehicle's engine and its systems.
- class [EngineIdleSoundComponent](#)
Sound of an engine idling.
- class [EngineStartStopComponent](#)
Sound of an engine starting / stopping. First audio clip is for engine starting, and second one is for engine stopping.
- class [ExhaustSmoke](#)
Controls particle emitters that represent exhausts.
- class [FlipOver](#)
- class [ForcedInduction](#)
Supercharger, turbocharger, etc. Can also be used on vehicles with no forced induction for sound effects such as intake noise or engine fan noise.
- class [Fuel](#)
- class [GearChangeComponent](#)
Shifter sound played when changing gears. Supports multiple audio clips of which one is chosen at random each time this effect is played.
- class [GroundDetection](#)
- class [HornComponent](#)
Vehicle horn sound.
- class [InputStates](#)
Class for storing input states of the vehicle.
- class [Lights](#)
Class for controlling all of the vehicle lights.
- class [Metrics](#)
Class for holding metrics such as odometer, top speed and drift time. All the speed values are in m/s. If you need the value in km/h or mph use [UnitConverter](#) functions.
- class [MobileInputManager](#)
Class for handling mobile user input via touch screen and sensors. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.

- class [RenderCameraDisable](#)
Attach this to any cameras rendering to mirrors so that when vehicle is inactive mirror will not update.
- class [Rigging](#)
- class [SkidComponent](#)
Sound produced by tire skidding over surface.
- class [SkidmarkDestroy](#)
Destroys skidmark object when distance to the vehicle becomes greater then distance threshold.
- class [SkidmarkGenerator](#)
Generates skidmark meshes.
- class [Skidmarks](#)
- class [Sound](#)
Main class that manages all the sound aspects of the vehicle.
- class [SoundComponent](#)
Base abstract class from which all vehicle sound components inherit.
- class [Stability](#)
Traction control class.
- class [Steering](#)
Everything related to steering and axle's geometry.
- class [SteeringWheel](#)
Script for controlling the GUI steering wheel for mobile controls. Credits go to yasirkula from Unity Forums for original code.
- class [SurfaceComponent](#)
Sounds produced by tire rolling over the surface.
- class [SurfaceParticles](#)
Skid smoke and surface dust generated by wheel slipping / rolling over the surface.
- class [SuspensionComponent](#)
Sound of wheel hitting the surface or obstracle.
- class [Tracks](#)
Class for handling tracked vehicles. If enabled all the wheels should be the same size and placed on either left or right side of the vehicle. Center wheels are not supported in this mode.
- class [TractionControl](#)
Traction control class.
- class [TrailerHandler](#)
Everthing related to a trailer.
- class [Transmission](#)
Handles gear changing and also torque and RPM transmission in both directions.
- class [TransmissionWhineComponent](#)
- class [TurboFlutterComponent](#)
Sound of wastegate releasing air on turbocharged vehicles.
- class [TurboWhistleComponent](#)
Sound of turbocharger or supercharger.
- class **UnitConverter**
- class [VehicleChanger](#)
Used for chaning vehicles. Also activates and deactivates vehicle cameras if default VehicleCamera system is used.
- class [VehicleController](#)
Main class controlling all the other parts of the vehicle.
- class [VehicleControllerInspector](#)
Shows different settings for trailer depending if vehicle is a trailer or a towing vehicle.
- class [VehicleFromScript](#)

Example script for setting up a vehicle at runtime - modify as/if needed. Sets up a vehicle from script at runtime. Model is required to already have body colliders set up, as well as wheels tagged with correct tag. Model also needs to have correct rotation (Z-forward, Y-up, X-right). Works only on vehicles with two wheels per axle. Number of axles is not limited.

- class [Wheel](#)

Contains everything related to wheels. To access WC3D's properties directly use WheelController getter. Most used variables are wrapped in getters and setter to enable eventual (but unlikely) future compatibility with default wheel collider. [Wheel](#) class is not equal to WheelController class. To access WC3D (WheelController) use WheelController getter/setter.

5.3 NWH.WheelController3D Namespace Reference

Classes

- class [WheelController](#)
API for [WheelController](#)
- class [WheelControllerInspector](#)

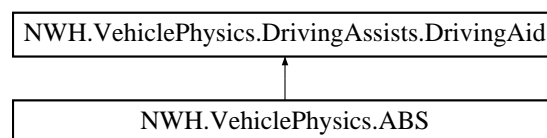
Chapter 6

Class Documentation

6.1 NWH.VehiclePhysics.ABS Class Reference

Traction control class.

Inheritance diagram for NWH.VehiclePhysics.ABS:



Public Member Functions

- void [Update](#) ([VehicleController](#) vc)

Additional Inherited Members

6.1.1 Detailed Description

Traction control class.

6.1.2 Member Function Documentation

6.1.2.1 Update()

```
void NWH.VehiclePhysics.ABS.Update (
    VehicleController vc )
```

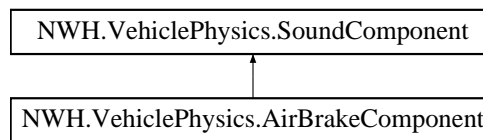
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/[ABS.cs](#)

6.2 NWH.VehiclePhysics.AirBrakeComponent Class Reference

Hiss produced by air brakes releasing air. Accepts multiple clips of which one will be chosen at random each time this effect is played.

Inheritance diagram for NWH.VehiclePhysics.AirBrakeComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, [AudioMixerGroup](#) amg)
- override void [Update](#) ()

Additional Inherited Members

6.2.1 Detailed Description

Hiss produced by air brakes releasing air. Accepts multiple clips of which one will be chosen at random each time this effect is played.

6.2.2 Member Function Documentation

6.2.2.1 Initialize()

```
override void NWH.VehiclePhysics.AirBrakeComponent.Initialize (
    VehicleController vc,
    AudioMixerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.2.2.2 Update()

```
override void NWH.VehiclePhysics.AirBrakeComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[AirBrakeComponent.cs](#)

6.3 NWH.VehiclePhysics.Axle Class Reference

Represents a single axle of a vehicle.

Classes

- class [Geometry](#)
Class holding all geometry related data for axle and it's wheels.

Public Types

- enum [DifferentialType](#) { [DifferentialType.Equal](#), [DifferentialType.Open](#), [DifferentialType.LimitedSlip](#), [DifferentialType.Locking](#) }
Axle differential. Equal - torque will be split equally between wheels at all times. Open - faster spinning wheel will receive more torque. Limited Slip - both wheels will always get some torque, depends on RPM of each wheel. Locking - slower spinning wheel will receive most torque.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()
- void [TorqueSplit](#) (float torque, float topRPM)
Splits torque between left and right wheel according to the differential setting. Not used with tracked vehicles.

Public Attributes

- [Wheel leftWheel](#) = new [Wheel](#)()
Axle's left wheel.
- [Wheel rightWheel](#) = new [Wheel](#)()
Axle's right wheel.
- [Geometry geometry](#) = new [Geometry](#)()
- float [powerCoefficient](#) = 1f
Amount of power that the axle will receive shown as a ratio. If two axles have both power coefficient of 1 each will receive half of total power (1:1), if first axle has p.c. of 1 and rear has p.c. of 0.5, this means that first axle will receive $(1 / (1 + 0.5)) = 0.66$ (66%) of total power and rear will receive $(0.5 / (1 + 0.5)) = 0.33$ (33%) of total power.
- float [brakeCoefficient](#) = 1f
If set to 1 axle will receive full brake torque as set by Max Torque parameter under Brake section while 0 means no breaking at all.
- float [handbrakeCoefficient](#)
If set to 1 axle will receive full brake torque when handbrake is used.
- float [differentialStrength](#)
Strength of the axle differential. Affects LimitedSlip and Locking differentials.
- [DifferentialType differentialType](#) = [DifferentialType.LimitedSlip](#)
Type of differential to be used when splitting torque between left and right wheel.

Properties

- float **Bias** [get, set]
- bool **IsPowered** [get]
Returns true if axle can receive power / torque.
- float **RPM** [get]
RPM of the axle as an average between wheels.
- float **SmoothRPM** [get]
Smoothed RPM of the axle.
- float **NoSlipRPM** [get]
RPM of the axle as if the wheels are not slipping.
- bool **WheelSpin** [get]
True if there is longitudinal slip on axles's left or right wheel.

6.3.1 Detailed Description

Represents a single axle of a vehicle.

6.3.2 Member Enumeration Documentation

6.3.2.1 DifferentialType

```
enum NWH.VehiclePhysics.Axle.DifferentialType [strong]
```

Axle differential. Equal - torque will be split equally between wheels at all times. Open - faster spinning wheel will receive more torque. Limited Slip - both wheels will always get some torque, depends on RPM of each wheel. Locking - slower spinning wheel will receive most torque.

Enumerator

Equal	
Open	
LimitedSlip	
Locking	

6.3.3 Member Function Documentation

6.3.3.1 Initialize()

```
void NWH.VehiclePhysics.Axle.Initialize (
    VehicleController vc )
```

6.3.3.2 TorqueSplit()

```
void NWH.VehiclePhysics.Axle.TorqueSplit (
    float torque,
    float topRPM )
```

Splits torque between left and right wheel according to the differential setting. Not used with tracked vehicles.

Parameters

<i>torque</i>	Amount of torque that will be split.
<i>topRPM</i>	Maximim RPM that axle is currently allowed to have.

6.3.3.3 Update()

```
void NWH.VehiclePhysics.Axle.Update ( )
```

6.3.4 Member Data Documentation

6.3.4.1 brakeCoefficient

```
float NWH.VehiclePhysics.Axle.brakeCoefficient = 1f
```

If set to 1 axle will receive full brake torque as set by Max Torque parameter under Brake section while 0 means no breaking at all.

6.3.4.2 differentialStrength

```
float NWH.VehiclePhysics.Axle.differentialStrength
```

Strength of the axle differential. Affects LimitedSlip and Locking differentials.

6.3.4.3 differentialType

```
DifferentialType NWH.VehiclePhysics.Axle.differentialType = DifferentialType.LimitedSlip
```

Type of differential to be used when splitting torque between left and right wheel.

6.3.4.4 geometry

```
Geometry NWH.VehiclePhysics.Axle.geometry = new Geometry()
```

6.3.4.5 handbrakeCoefficient

```
float NWH.VehiclePhysics.Axle.handbrakeCoefficient
```

If set to 1 axle will receive full brake torque when handbrake is used.

6.3.4.6 leftWheel

```
Wheel NWH.VehiclePhysics.Axle.leftWheel = new Wheel()
```

Axle's left wheel.

6.3.4.7 powerCoefficient

```
float NWH.VehiclePhysics.Axle.powerCoefficient = 1f
```

Amount of power that the axle will receive shown as a ratio. If two axles have both power coefficient of 1 each will receive half of total power (1:1), if first axle has p.c. of 1 and rear has p.c. of 0.5, this means that first axle will receive $(1 / (1 + 0.5)) = 0.66$ (66%) of total power and rear will receive $(0.5 / (1 + 0.5)) = 0.33$ (33%) of total power.

6.3.4.8 rightWheel

```
Wheel NWH.VehiclePhysics.Axle.rightWheel = new Wheel()
```

Axle's right wheel.

6.3.5 Property Documentation

6.3.5.1 Bias

```
float NWH.VehiclePhysics.Axle.Bias [get], [set]
```

6.3.5.2 IsPowered

```
bool NWH.VehiclePhysics.Axle.IsPowered [get]
```

Returns true if axle can receive power / torque.

6.3.5.3 NoSlipRPM

```
float NWH.VehiclePhysics.Axle.NoSlipRPM [get]
```

RPM of the axle as if the wheels are not slipping.

6.3.5.4 RPM

```
float NWH.VehiclePhysics.Axle.RPM [get]
```

RPM of the axle as an average between wheels.

6.3.5.5 SmoothRPM

```
float NWH.VehiclePhysics.Axle.SmoothRPM [get]
```

Smoothed RPM of the axle.

6.3.5.6 WheelSpin

```
bool NWH.VehiclePhysics.Axle.WheelSpin [get]
```

True if there is longitudinal slip on axles's left or right wheel.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Axle.cs](#)

6.4 NWH.VehiclePhysics.Backfire Class Reference

Controls exhaust flames / flashes.

Public Member Functions

- void `Initialize` (`VehicleController` vc)
- void `Update` ()
- void `Flash` ()

Public Attributes

- float `duration` = 0.05f
- List< `MeshRenderer` > `renderers` = new List<MeshRenderer>()

6.4.1 Detailed Description

Controls exhaust flames / flashes.

6.4.2 Member Function Documentation

6.4.2.1 `Flash()`

```
void NWH.VehiclePhysics.Backfire.Flash ( )
```

6.4.2.2 `Initialize()`

```
void NWH.VehiclePhysics.Backfire.Initialize (
    VehicleController vc )
```

6.4.2.3 `Update()`

```
void NWH.VehiclePhysics.Backfire.Update ( )
```

6.4.3 Member Data Documentation

6.4.3.1 `duration`

```
float NWH.VehiclePhysics.Backfire.duration = 0.05f
```


6.4.3.2 renderers

```
List<MeshRenderer> NWH.VehiclePhysics.Backfire.renderers = new List<MeshRenderer>()
```

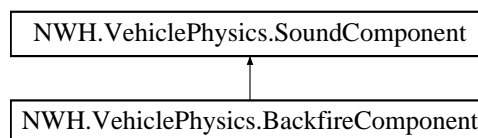
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/[Backfire.cs](#)

6.5 NWH.VehiclePhysics.BackfireComponent Class Reference

Exhaust popping on deceleration / rev limiter.

Inheritance diagram for NWH.VehiclePhysics.BackfireComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, AudioManagerGroup amg)
- override void [Update](#) ()

Additional Inherited Members

6.5.1 Detailed Description

Exhaust popping on deceleration / rev limiter.

6.5.2 Member Function Documentation

6.5.2.1 Initialize()

```
override void NWH.VehiclePhysics.BackfireComponent.Initialize (  
    VehicleController vc,  
    AudioManagerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.5.2.2 Update()

```
override void NWH.VehiclePhysics.BackfireComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

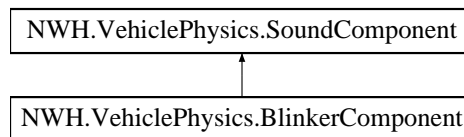
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[BackfireComponent.cs](#)

6.6 NWH.VehiclePhysics.BlinkerComponent Class Reference

Click-clack of the working blinker. Accepts two clips, first is for the blinker turning on and the second is for blinker turning off.

Inheritance diagram for NWH.VehiclePhysics.BlinkerComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, AudioManagerGroup amg)
- override void [Update](#) ()

Additional Inherited Members

6.6.1 Detailed Description

Click-clack of the working blinker. Accepts two clips, first is for the blinker turning on and the second is for blinker turning off.

6.6.2 Member Function Documentation

6.6.2.1 Initialize()

```
override void NWH.VehiclePhysics.BlinkerComponent.Initialize (
    VehicleController vc,
    AudioManagerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.6.2.2 Update()

```
override void NWH.VehiclePhysics.BlinkerComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[BlinkerComponent.cs](#)

6.7 NWH.VehiclePhysics.Brakes Class Reference

Public Member Functions

- void [Update](#) ([VehicleController](#) vc)

Public Attributes

- float [maxTorque](#) = 5000f
Max brake torque that can be applied to each wheel. To adjust braking on per-axle basis change brake coefficients under [Axle](#) settings.
- float [frictionTorque](#) = 120f
Imitation of rolling resistance and friction between drivetrain parts. Applied to all wheels.
- float [smoothing](#) = 0.9f
Time in seconds needed to reach full braking torque.
- bool [brakeWhileIdle](#) = true
If true vehicle will break when in neutral and no throttle is applied.
- bool [airBrakes](#) = false
Set to true to use the air brake sound effect.
- float [airBrakePressure](#)

Properties

- bool [Active](#) [get, set]
Retruns true if vehicle is currently braking. Will return true if there is ANY brake torque applied to the wheels.

6.7.1 Member Function Documentation

6.7.1.1 Update()

```
void NWH.VehiclePhysics.Brakes.Update (
    VehicleController vc )
```

6.7.2 Member Data Documentation

6.7.2.1 airBrakePressure

```
float NWH.VehiclePhysics.Brakes.airBrakePressure
```

6.7.2.2 airBrakes

```
bool NWH.VehiclePhysics.Brakes.airBrakes = false
```

Set to true to use the air brake sound effect.

6.7.2.3 brakeWhileIdle

```
bool NWH.VehiclePhysics.Brakes.brakeWhileIdle = true
```

If true vehicle will break when in neutral and no throttle is applied.

6.7.2.4 frictionTorque

```
float NWH.VehiclePhysics.Brakes.frictionTorque = 120f
```

Imitation of rolling resistance and friction between drivetrain parts. Applied to all wheels.

6.7.2.5 maxTorque

```
float NWH.VehiclePhysics.Brakes.maxTorque = 5000f
```

Max brake torque that can be applied to each wheel. To adjust braking on per-axle basis change brake coefficients under [Axle](#) settings.

6.7.2.6 smoothing

```
float NWH.VehiclePhysics.Brakes.smoothing = 0.9f
```

Time in seconds needed to reach full braking torque.

6.7.3 Property Documentation

6.7.3.1 Active

```
bool NWH.VehiclePhysics.Brakes.Active [get], [set]
```

Retruns true if vehicle is currently braking. Will return true if there is ANY brake torque applied to the wheels.

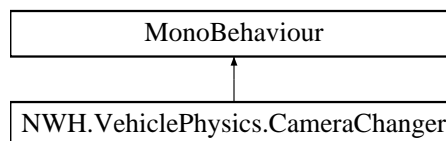
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Brakes.cs](#)

6.8 NWH.VehiclePhysics.CameraChanger Class Reference

Switches between the camera objects that are children to this object and contain camera tag, in order they appear in the hierarchy or in order they are added to the vehicle cameras list.

Inheritance diagram for NWH.VehiclePhysics.CameraChanger:



Public Member Functions

- void [NextCamera](#) ()
Activates next camera in order the camera scripts are attached to the camera object.

Public Attributes

- int [currentCamera](#) = 0
Index of the camera from vehicle cameras list that will be active first.
- string [cameraTag](#) = "VehicleCamera"
Cameras with this tag will be added to the vehicle cameras list. Cameras need to be children of this object.
- List< GameObject > [vehicleCameras](#) = new List<GameObject>()
List of cameras that the changer will cycle through. Leave empty if you want cameras to be automatically detected. To be detected cameras need to have camera tag and be children of the object this script is attached to.

6.8.1 Detailed Description

Switches between the camera objects that are children to this object and contain camera tag, in order they appear in the hierarchy or in order they are added to the vehicle cameras list.

6.8.2 Member Function Documentation

6.8.2.1 NextCamera()

```
void NWH.VehiclePhysics.CameraChanger.NextCamera ( )
```

Activates next camera in order the camera scripts are attached to the camera object.

6.8.3 Member Data Documentation

6.8.3.1 cameraTag

```
string NWH.VehiclePhysics.CameraChanger.cameraTag = "VehicleCamera"
```

Cameras with this tag will be added to the vehicle cameras list. Cameras need to be children of this object.

6.8.3.2 currentCamera

```
int NWH.VehiclePhysics.CameraChanger.currentCamera = 0
```

Index of the camera from vehicle cameras list that will be active first.

6.8.3.3 vehicleCameras

```
List<GameObject> NWH.VehiclePhysics.CameraChanger.vehicleCameras = new List<GameObject>()
```

List of cameras that the changer will cycle through. Leave empty if you want cameras to be automatically detected. To be detected cameras need to have camera tag and be children of the object this script is attached to.

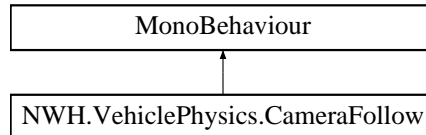
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/[CameraChanger.cs](#)

6.9 NWH.VehiclePhysics.CameraFollow Class Reference

Camera that follows behind the vehicle.

Inheritance diagram for NWH.VehiclePhysics.CameraFollow:



Static Public Member Functions

- static float [AngleSigned](#) (Vector3 v1, Vector3 v2, Vector3 n)
Determine the signed angle between two vectors, with normal 'n' as the rotation axis.

Public Attributes

- Transform [target](#)
Target transform that the camera will follow.
- float [distance](#) = 6f
Distance at which camera will follow.
- float [height](#) = 2.5f
Height in relation to the target at which the camera will follow.
- float [targetUpOffset](#) = 1.25f
Offset in the up direction from the target. Use this if you do not want to use camera baits.
- float [targetForwardOffset](#) = 0
Offset in the forward direction from the target. Use this if you do not want to use camera baits.
- float [smoothing](#) = 0.2f
Positional and rotational smoothing of the camera.
- float [angleFollowStrength](#) = 0
Allows camera to match target's angle to some extent.

6.9.1 Detailed Description

Camera that follows behind the vehicle.

6.9.2 Member Function Documentation

6.9.2.1 AngleSigned()

```
static float NWH.VehiclePhysics.CameraFollow.AngleSigned (
    Vector3 v1,
    Vector3 v2,
    Vector3 n ) [static]
```

Determine the signed angle between two vectors, with normal 'n' as the rotation axis.

6.9.3 Member Data Documentation

6.9.3.1 angleFollowStrength

```
float NWH.VehiclePhysics.CameraFollow.angleFollowStrength = 0
```

Allows camera to match target's angle to some extent.

6.9.3.2 distance

```
float NWH.VehiclePhysics.CameraFollow.distance = 6f
```

Distance at which camera will follow.

6.9.3.3 height

```
float NWH.VehiclePhysics.CameraFollow.height = 2.5f
```

Height in relation to the target at which the camera will follow.

6.9.3.4 smoothing

```
float NWH.VehiclePhysics.CameraFollow.smoothing = 0.2f
```

Positional and rotational smoothing of the camera.

6.9.3.5 target

```
Transform NWH.VehiclePhysics.CameraFollow.target
```

Target transform that the camera will follow.

6.9.3.6 targetForwardOffset

```
float NWH.VehiclePhysics.CameraFollow.targetForwardOffset = 0
```

Offset in the forward direction from the target. Use this if you do not want to use camera baits.

6.9.3.7 targetUpOffset

```
float NWH.VehiclePhysics.CameraFollow.targetUpOffset = 1.25f
```

Offset in the up direction from the target. Use this if you do not want to use camera baits.

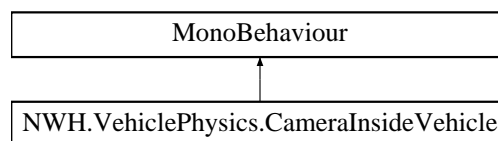
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/[CameraFollow.cs](#)

6.10 NWH.VehiclePhysics.CameraInsideVehicle Class Reference

Empty component that should be attached to the cameras that are inside the vehicle if interior sound change is to be used.

Inheritance diagram for NWH.VehiclePhysics.CameraInsideVehicle:



6.10.1 Detailed Description

Empty component that should be attached to the cameras that are inside the vehicle if interior sound change is to be used.

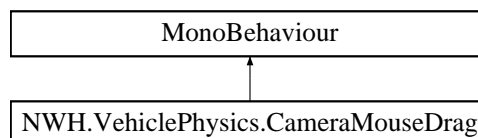
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/[CameraInsideVehicle.cs](#)

6.11 NWH.VehiclePhysics.CameraMouseDrag Class Reference

Camera that can be dragged with the mouse.

Inheritance diagram for NWH.VehiclePhysics.CameraMouseDrag:



Public Member Functions

- float [ClampAngle](#) (float angle, float min, float max)

Public Attributes

- Transform [target](#)
Target at which the camera will look.
- float [distance](#)
Distance from target at which camera will be positioned. Might vary depending on smoothing.
- float [minDistance](#) = 5.0f
Minimum distance that will be reached when zooming in.
- float [maxDistance](#) = 13.0f
Maximum distance that will be reached when zooming out.
- float [horizontalMouseSensitivity](#) = 5.0f
- float [verticalMouseSensitivity](#) = 5.0f
- float [mouseWheelSensitivity](#) = 5.0f
- float [verticalMinAngle](#) = -40.0f
- float [verticalMaxAngle](#) = 80.0f
- float [smoothing](#) = 0.05f
- float [distanceSmoothing](#) = 0.05f
- bool [followTargetsRotation](#) = false

6.11.1 Detailed Description

Camera that can be dragged with the mouse.

6.11.2 Member Function Documentation

6.11.2.1 ClampAngle()

```
float NWH.VehiclePhysics.CameraMouseDrag.ClampAngle (
    float angle,
    float min,
    float max )
```

6.11.3 Member Data Documentation

6.11.3.1 distance

```
float NWH.VehiclePhysics.CameraMouseDrag.distance
```

Distance from target at which camera will be positioned. Might vary depending on smoothing.

6.11.3.2 distanceSmoothing

```
float NWH.VehiclePhysics.CameraMouseDrag.distanceSmoothing = 0.05f
```

6.11.3.3 followTargetsRotation

```
bool NWH.VehiclePhysics.CameraMouseDrag.followTargetsRotation = false
```

6.11.3.4 horizontalMouseSensitivity

```
float NWH.VehiclePhysics.CameraMouseDrag.horizontalMouseSensitivity = 5.0f
```

6.11.3.5 maxDistance

```
float NWH.VehiclePhysics.CameraMouseDrag.maxDistance = 13.0f
```

Maximum distance that will be reached when zooming out.

6.11.3.6 minDistance

```
float NWH.VehiclePhysics.CameraMouseDrag.minDistance = 5.0f
```

Minimum distance that will be reached when zooming in.

6.11.3.7 mouseWheelSensitivity

```
float NWH.VehiclePhysics.CameraMouseDrag.mouseWheelSensitivity = 5.0f
```

6.11.3.8 smoothing

```
float NWH.VehiclePhysics.CameraMouseDrag.smoothing = 0.05f
```

6.11.3.9 target

`Transform NWH.VehiclePhysics.CameraMouseDrag.target`

Target at which the camera will look.

6.11.3.10 verticalMaxAngle

`float NWH.VehiclePhysics.CameraMouseDrag.verticalMaxAngle = 80.0f`

6.11.3.11 verticalMinAngle

`float NWH.VehiclePhysics.CameraMouseDrag.verticalMinAngle = -40.0f`

6.11.3.12 verticalMouseSensitivity

`float NWH.VehiclePhysics.CameraMouseDrag.verticalMouseSensitivity = 5.0f`

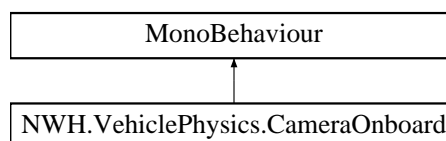
The documentation for this class was generated from the following file:

- `E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraMouseDrag.cs`

6.12 NWH.VehiclePhysics.CameraOnboard Class Reference

Camera for on or in-vehicle use with option of head movement according to the G-forces.

Inheritance diagram for `NWH.VehiclePhysics.CameraOnboard`:



Public Attributes

- `VehicleController vehicleController`
Vehicle Controller that this script is targeting. Can be left empty if head movement is not being used.
- `float positionSmoothing = 0.3f`
Smoothing of the head movement.
- `float positionIntensity = 0.125f`
How much will the head move around for the given g-force.
- `float maxPositionOffsetMagnitude = 0.2f`
Maximum head movement from the initial position.

6.12.1 Detailed Description

Camera for on or in-vehicle use with option of head movement according to the G-forces.

6.12.2 Member Data Documentation

6.12.2.1 maxPositionOffsetMagnitude

```
float NWH.VehiclePhysics.CameraOnboard.maxPositionOffsetMagnitude = 0.2f
```

Maximum head movement from the initial position.

6.12.2.2 positionIntensity

```
float NWH.VehiclePhysics.CameraOnboard.positionIntensity = 0.125f
```

How much will the head move around for the given g-force.

6.12.2.3 positionSmoothing

```
float NWH.VehiclePhysics.CameraOnboard.positionSmoothing = 0.3f
```

Smoothing of the head movement.

6.12.2.4 vehicleController

```
VehicleController NWH.VehiclePhysics.CameraOnboard.vehicleController
```

Vehicle Controller that this script is targeting. Can be left empty if head movement is not being used.

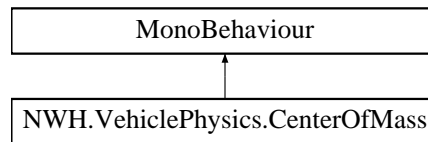
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/[CameraOnboard.cs](#)

6.13 NWH.VehiclePhysics.CenterOfMass Class Reference

Used for adjusting center of mass of any rigidbody object.

Inheritance diagram for NWH.VehiclePhysics.CenterOfMass:



Public Attributes

- Vector3 `centerOfMassOffset` = Vector3.zero
Center of mass offset in relation to the original COM. Needs to be readjusted when new colliders are added.
- bool `showCOM` = true
Enable to show a green spehere at the current center of mass.

6.13.1 Detailed Description

Used for adjusting center of mass of any rigidbody object.

6.13.2 Member Data Documentation

6.13.2.1 centerOfMassOffset

```
Vector3 NWH.VehiclePhysics.CenterOfMass.centerOfMassOffset = Vector3.zero
```

Center of mass offset in relation to the original COM. Needs to be readjusted when new colliders are added.

6.13.2.2 showCOM

```
bool NWH.VehiclePhysics.CenterOfMass.showCOM = true
```

Enable to show a green spehere at the current center of mass.

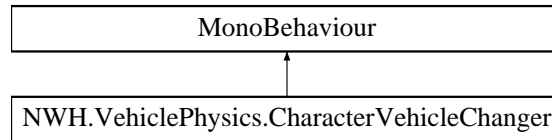
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/[CenterOfMass.cs](#)

6.14 NWH.VehiclePhysics.CharacterVehicleChanger Class Reference

Allows character to enter or exit vehicle. Can be used with any first or 3rd person object.

Inheritance diagram for NWH.VehiclePhysics.CharacterVehicleChanger:



Public Attributes

- float `enterDistance` = 2f
Maximum distance at which the character will be able to enter the vehicle.
- float `maxEnterExitVehicleSpeed` = 2f
Maximum speed at which the character will be able to enter / exit the vehicle.
- string `enterExitTag` = "EnterExitPoint"
Tag of the object representing the point from which the enter distance will be measured. Useful if you want to enable you character to enter only when near the door.
- GameObject `characterControllerObject`
Game object representing a character. Can also be another vehicle.
- bool `nearVehicle` = false
True when character can enter the vehicle.

6.14.1 Detailed Description

Allows character to enter or exit vehicle. Can be used with any first or 3rd person object.

6.14.2 Member Data Documentation

6.14.2.1 characterControllerObject

```
GameObject NWH.VehiclePhysics.CharacterVehicleChanger.characterControllerObject
```

Game object representing a character. Can also be another vehicle.

6.14.2.2 enterDistance

```
float NWH.VehiclePhysics.CharacterVehicleChanger.enterDistance = 2f
```

Maximum distance at which the character will be able to enter the vehicle.

6.14.2.3 enterExitTag

```
string NWH.VehiclePhysics.CharacterVehicleChanger.enterExitTag = "EnterExitPoint"
```

Tag of the object representing the point from which the enter distance will be measured. Useful if you want to enable you character to enter only when near the door.

6.14.2.4 maxEnterExitVehicleSpeed

```
float NWH.VehiclePhysics.CharacterVehicleChanger.maxEnterExitVehicleSpeed = 2f
```

Maximum speed at which the character will be able to enter / exit the vehicle.

6.14.2.5 nearVehicle

```
bool NWH.VehiclePhysics.CharacterVehicleChanger.nearVehicle = false
```

True when character can enter the vehicle.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/[CharacterVehicleChanger.cs](#)

6.15 NWH.VehiclePhysics.DamageHandler.CollisionEvent Class Reference

Contains data on the collision that has last happened.

Public Attributes

- Queue< MeshFilter > [deformationQueue](#) = new Queue<MeshFilter>()
Queue of mesh filter components that are waiting for deformation. Some of the meshes might be queued for checking even if not deformed.
- Collision [collision](#)
Collision data for the collision event.
- float [decelerationMagnitude](#)
Magnitude of the deceleration vector at the moment of impact.

6.15.1 Detailed Description

Contains data on the collision that has last happened.

6.15.2 Member Data Documentation

6.15.2.1 collision

`Collision NWH.VehiclePhysics.DamageHandler.CollisionEvent.collision`

Collision data for the collision event.

6.15.2.2 decelerationMagnitude

`float NWH.VehiclePhysics.DamageHandler.CollisionEvent.decelerationMagnitude`

Magnitude of the deceleration vector at the moment of impact.

6.15.2.3 deformationQueue

```
Queue<MeshFilter> NWH.VehiclePhysics.DamageHandler.CollisionEvent.deformationQueue = new
Queue<MeshFilter> ()
```

Queue of mesh filter components that are waiting for deformation. Some of the meshes might be queued for checking even if not deformed.

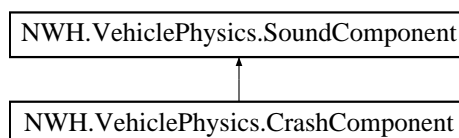
The documentation for this class was generated from the following file:

- `E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/`[DamageHandler.cs](#)

6.16 NWH.VehiclePhysics.CrashComponent Class Reference

[Sound](#) of vehicle crashing into an object. Supports multiple audio clips of which one will be chosen at random each time this effect is played.

Inheritance diagram for NWH.VehiclePhysics.CrashComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, AudioManagerGroup amg)
- override void [Update](#) ()
- void [Play](#) ([VehicleController](#) vc)

Additional Inherited Members

6.16.1 Detailed Description

[Sound](#) of vehicle crashing into an object. Supports multiple audio clips of which one will be chosen at random each time this effect is played.

6.16.2 Member Function Documentation

6.16.2.1 Initialize()

```
override void NWH.VehiclePhysics.CrashComponent.Initialize (
    VehicleController vc,
    AudioManagerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.16.2.2 Play()

```
void NWH.VehiclePhysics.CrashComponent.Play (
    VehicleController vc )
```

6.16.2.3 Update()

```
override void NWH.VehiclePhysics.CrashComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[CrashComponent.cs](#)

6.17 NWH.VehiclePhysics.DamageHandler Class Reference

Handles all damage related calculations and mesh deformations. Collision sounds are handled by [CrashComponent](#) class.

Classes

- class [CollisionEvent](#)
Contains data on the collision that has last happened.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()
- void [Enqueue](#) (Collision collision, float accelerationMagnitude)
Add collision to the queue of collisions waiting to be processed.
- void [MeshDeform](#) ([CollisionEvent](#) collisionEvent, [MeshFilter](#) deformableMeshFilter)
Deforms a mesh using data from collision event.

Public Attributes

- bool [enabled](#) = false
Determines if damage, mesh deformation and performance degradation will be used.
- bool [performanceDegradation](#) = false
Should damage affect vehicle performance (steering, power, etc.)?
- float [allowedDamage](#) = 50000
Maximum allowed damage before the vehicle breaks down. Performance will decline as damage is nearing allowed damage.
- int [deformationVerticesPerFrame](#) = 8000
Number of vertices that will be checked and eventually deformed per frame.
- float [deformationRadius](#) = 0.6f
Radius is which vertices will be deformed.
- float [deformationStrength](#) = 1.6f
Determines how much vertices will be deformed for given collision strength.
- float [deformationRandomness](#) = 0.1f
Adds noise to the mesh deformation. 0 will result in smooth mesh.
- float [decelerationThreshold](#) = 30f
Deceleration magnitude needed to trigger damage.
- List< string > [ignoreTags](#) = new List<string>()
Collisions with the objects that have a tag that is on this list will be ignored. Collision state will be changed but no processing will happen.
- int [previousCollisionHash](#)
Hash of the previous queued collision. Prevents reacting to the same collision twice since collision is called during OnCollisionStay() so more data can be collected.

Properties

- float [Damage](#) [get, set]
Current vehicle damage.
- float [DamagePercent](#) [get]
Current vehicle damage. Percentage from allowed damage.

6.17.1 Detailed Description

Handles all damage related calculations and mesh deformations. Collision sounds are handled by [CrashComponent](#) class.

6.17.2 Member Function Documentation

6.17.2.1 Enqueue()

```
void NWH.VehiclePhysics.DamageHandler.Enqueue (
    Collision collision,
    float accelerationMagnitude )
```

Add collision to the queue of collisions waiting to be processed.

6.17.2.2 Initialize()

```
void NWH.VehiclePhysics.DamageHandler.Initialize (
    VehicleController vc )
```

6.17.2.3 MeshDeform()

```
void NWH.VehiclePhysics.DamageHandler.MeshDeform (
    CollisionEvent collisionEvent,
    MeshFilter deformableMeshFilter )
```

Deforms a mesh using data from collision event.

6.17.2.4 Update()

```
void NWH.VehiclePhysics.DamageHandler.Update ( )
```

6.17.3 Member Data Documentation

6.17.3.1 allowedDamage

```
float NWH.VehiclePhysics.DamageHandler.allowedDamage = 50000
```

Maximum allowed damage before the vehicle breaks down. Performance will decline as damage is nearing allowed damage.

6.17.3.2 decelerationThreshold

```
float NWH.VehiclePhysics.DamageHandler.decelerationThreshold = 30f
```

Deceleration magnitude needed to trigger damage.

6.17.3.3 deformationRadius

```
float NWH.VehiclePhysics.DamageHandler.deformationRadius = 0.6f
```

Radius is which vertices will be deformed.

6.17.3.4 deformationRandomness

```
float NWH.VehiclePhysics.DamageHandler.deformationRandomness = 0.1f
```

Adds noise to the mesh deformation. 0 will result in smooth mesh.

6.17.3.5 deformationStrength

```
float NWH.VehiclePhysics.DamageHandler.deformationStrength = 1.6f
```

Determines how much vertices will be deformed for given collision strength.

6.17.3.6 deformationVerticesPerFrame

```
int NWH.VehiclePhysics.DamageHandler.deformationVerticesPerFrame = 8000
```

Number of vertices that will be checked and eventually deformed per frame.

6.17.3.7 enabled

```
bool NWH.VehiclePhysics.DamageHandler.enabled = false
```

Determines if damage, mesh deformation and performance degradation will be used.

6.17.3.8 ignoreTags

```
List<string> NWH.VehiclePhysics.DamageHandler.ignoreTags = new List<string>()
```

Collisions with the objects that have a tag that is on this list will be ignored. Collision state will be changed but no processing will happen.

6.17.3.9 performanceDegradation

```
bool NWH.VehiclePhysics.DamageHandler.performanceDegradation = false
```

Should damage affect vehicle performance (steering, power, etc.)?

6.17.3.10 previousCollisionHash

```
int NWH.VehiclePhysics.DamageHandler.previousCollisionHash
```

Hash of the previous queued collision. Prevents reacting to the same collision twice since collision is called during OnCollisionStay() so more data can be collected.

6.17.4 Property Documentation

6.17.4.1 Damage

```
float NWH.VehiclePhysics.DamageHandler.Damage [get], [set]
```

Current vehicle damage.

6.17.4.2 DamagePercent

```
float NWH.VehiclePhysics.DamageHandler.DamagePercent [get]
```

Current vehicle damage. Percentage from allowed damage.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[DamageHandler.cs](#)

6.18 NWH.WheelController3D.WheelController.Damper Class Reference

Suspension part.

Public Attributes

- AnimationCurve [dampingCurve](#) = null
- float [unitBumpForce](#) = 800.0f
- float [unitReboundForce](#) = 1000.0f
- float [force](#)
- float [maxForce](#)

6.18.1 Detailed Description

Suspension part.

6.18.2 Member Data Documentation

6.18.2.1 dampingCurve

```
AnimationCurve NWH.WheelController3D.WheelController.Damper.dampingCurve = null
```

6.18.2.2 force

```
float NWH.WheelController3D.WheelController.Damper.force
```

6.18.2.3 maxForce

```
float NWH.WheelController3D.WheelController.Damper.maxForce
```

6.18.2.4 unitBumpForce

```
float NWH.WheelController3D.WheelController.Damper.unitBumpForce = 800.0f
```

6.18.2.5 unitReboundForce

```
float NWH.WheelController3D.WheelController.Damper.unitReboundForce = 1000.0f
```

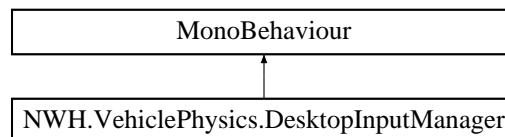
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/[WheelController.cs](#)

6.19 NWH.VehiclePhysics.DesktopInputManager Class Reference

Class for handling desktop user input via mouse and keyboard. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.

Inheritance diagram for NWH.VehiclePhysics.DesktopInputManager:



Public Types

- enum [InputType](#) { [InputType.Standard](#), [InputType.Mouse](#), [InputType.MouseSteer](#) }
Type of input user input. Standard - standard keyboard, joystick or gamepad input mapped through the input manager. Mouse - uses mouse position on screen to control throttle/braking and steering. MouseSteer - uses LMB / RMB for throttle and braking and mouse for steering.

Public Attributes

- [InputType](#) [inputType](#) = [InputType.Standard](#)
- [VehicleChanger](#) [vehicleChanger](#)
Set to null (none) if you want to use your own vehicle controller. If this is set to other than null current active vehicle according to the assigned vehicle changer will be used instead of the assigned vehicle controller.
- [VehicleController](#) [vehicleController](#)
If you want to use this script with a single vehicle or want to set your own vehicle controller from script set vehicle changer field to null / none.

6.19.1 Detailed Description

Class for handling desktop user input via mouse and keyboard. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.

6.19.2 Member Enumeration Documentation

6.19.2.1 InputType

```
enum NWH.VehiclePhysics.DesktopInputManager.InputType [strong]
```

Type of input user input. Standard - standard keyboard, joystick or gamepad input mapped through the input manager. Mouse - uses mouse position on screen to control throttle/braking and steering. MouseSteer - uses LMB / RMB for throttle and braking and mouse for steering.

Enumerator

Standard	
Mouse	
MouseSteer	

6.19.3 Member Data Documentation

6.19.3.1 inputType

```
InputType NWH.VehiclePhysics.DesktopInputManager.inputType = InputType.Standard
```

6.19.3.2 vehicleChanger

```
VehicleChanger NWH.VehiclePhysics.DesktopInputManager.vehicleChanger
```

Set to null (none) if you want to use your own vehicle controller. If this is set to other than null current active vehicle according to the assigned vehicle changer will be used instead of the assigned vehicle controller.

6.19.3.3 vehicleController

`VehicleController` `NWH.VehiclePhysics.DesktopInputManager.vehicleController`

If you want to use this script with a single vehicle or want to set your own vehicle controller from script set vehicle changer field to null / none.

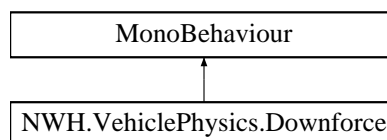
The documentation for this class was generated from the following file:

- `E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/DesktopInputManager.cs`

6.20 NWH.VehiclePhysics.Downforce Class Reference

Loose approximation on downforce acting on a vehicle. [Downforce](#) in this case is only dependent on speed, but its amount and speed at which it is achieved can be adjusted. Can be adjusted at runtime.

Inheritance diagram for `NWH.VehiclePhysics.Downforce`:



Classes

- class [DownforcePoint](#)
Single point at which downforce will be applied.

Public Attributes

- `List< DownforcePoint > downforcePoints = new List<DownforcePoint>()`
- `float maxDownforceSpeed = 50f`
Speed in m/s at which maximum downforce will be applied. Amount of downforce will grow exponentially up to this value.

6.20.1 Detailed Description

Loose approximation on downforce acting on a vehicle. [Downforce](#) in this case is only dependent on speed, but its amount and speed at which it is achieved can be adjusted. Can be adjusted at runtime.

6.20.2 Member Data Documentation

6.20.2.1 downforcePoints

```
List<DownforcePoint> NWH.VehiclePhysics.Downforce.downforcePoints = new List<DownforcePoint>()
```

6.20.2.2 maxDownforceSpeed

```
float NWH.VehiclePhysics.Downforce.maxDownforceSpeed = 50f
```

Speed in m/s at which maximum downforce will be applied. Amount of downforce will grow exponentially up to this value.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/[Downforce.cs](#)

6.21 NWH.VehiclePhysics.Downforce.DownforcePoint Class Reference

Single point at which downforce will be applied.

Public Attributes

- Vector3 [position](#)
Position relative to the vehicle at which downforce will be applied. Marked by red arrow gizmo.
- float [maxForce](#)
Maximim force in [N] that can be applied as a result of downforce. Putting in a too large value will hammer the vehicle into the ground since springs get overloaded.

6.21.1 Detailed Description

Single point at which downforce will be applied.

6.21.2 Member Data Documentation

6.21.2.1 maxForce

```
float NWH.VehiclePhysics.Downforce.DownforcePoint.maxForce
```

Maximim force in [N] that can be applied as a result of downforce. Putting in a too large value will hammer the vehicle into the ground since springs get overloaded.

6.21.2.2 position

```
Vector3 NWH.VehiclePhysics.Downforce.DownforcePoint.position
```

Position relative to the vehicle at which downforce will be applied. Marked by red arrow gizmo.

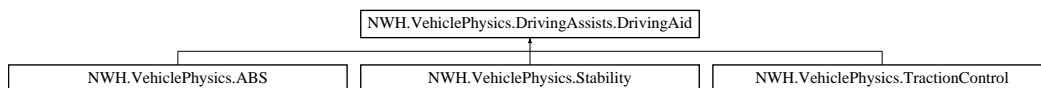
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/[Downforce.cs](#)

6.22 NWH.VehiclePhysics.DrivingAssists.DrivingAid Class Reference

Base class for driving aids.

Inheritance diagram for NWH.VehiclePhysics.DrivingAssists.DrivingAid:



Public Attributes

- bool [enabled](#) = false
Determines if driving aid should be used.
- bool [active](#)
Is driving aid currently active?
- float [intensity](#) = 0.3f
Higher intensity will result in driving aid affecting the vehicle behavior stronger.

6.22.1 Detailed Description

Base class for driving aids.

6.22.2 Member Data Documentation

6.22.2.1 active

```
bool NWH.VehiclePhysics.DrivingAssists.DrivingAid.active
```

Is driving aid currently active?

6.22.2.2 enabled

```
bool NWH.VehiclePhysics.DrivingAssists.DrivingAid.enabled = false
```

Determines if driving aid should be used.

6.22.2.3 intensity

```
float NWH.VehiclePhysics.DrivingAssists.DrivingAid.intensity = 0.3f
```

Higher intensity will result in driving aid affecting the vehicle behavior stronger.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/[DrivingAssists.cs](#)

6.23 NWH.VehiclePhysics.DrivingAssists Class Reference

Systems to help driver with vehicle control.

Classes

- class [DrivingAid](#)
Base class for driving aids.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()

Public Attributes

- [ABS](#) [abs](#) = new [ABS](#)()
Anti brake lock system.
- [TractionControl](#) [tcs](#) = new [TractionControl](#)()
Traction control.
- [Stability](#) [stability](#) = new [Stability](#)()
[Stability](#) help. It will improve vehicle handling so should not be used as an option in competitive games but rather to achieve more arcade-like behavior when needed. Force is applied that is not dependent on slip so vehicle will be able to steer even when skidding.

6.23.1 Detailed Description

Systems to help driver with vehicle control.

6.23.2 Member Function Documentation

6.23.2.1 Initialize()

```
void NWH.VehiclePhysics.DrivingAssists.Initialize (
    VehicleController vc )
```

6.23.2.2 Update()

```
void NWH.VehiclePhysics.DrivingAssists.Update ( )
```

6.23.3 Member Data Documentation

6.23.3.1 abs

```
ABS NWH.VehiclePhysics.DrivingAssists.abs = new ABS()
```

Anti brake lock system.

6.23.3.2 stability

```
Stability NWH.VehiclePhysics.DrivingAssists.stability = new Stability()
```

[Stability](#) help. It will improve vehicle handling so should not be used as an option in competitive games but rather to achieve more arcade-like behavior when needed. Force is applied that is not dependent on slip so vehicle will be able to steer even when skidding.

6.23.3.3 tcs

```
TractionControl NWH.VehiclePhysics.DrivingAssists.tcs = new TractionControl()
```

Traction control.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/[DrivingAssists.cs](#)

6.24 NWH.VehiclePhysics.Effects Class Reference

Main class for handling visual effects such as skidmarks, lights and exhausts.

Public Member Functions

- void `Initialize` (`VehicleController` vc)
- void `Update` ()

Public Attributes

- `Skidmarks skidmarks` = new `Skidmarks`()
- `Lights lights` = new `Lights`()
- `ExhaustSmoke exhausts` = new `ExhaustSmoke`()
- `Backfire exhaustFlash` = new `Backfire`()

6.24.1 Detailed Description

Main class for handling visual effects such as skidmarks, lights and exhausts.

6.24.2 Member Function Documentation

6.24.2.1 `Initialize()`

```
void NWH.VehiclePhysics.Effects.Initialize (  
    VehicleController vc )
```

6.24.2.2 `Update()`

```
void NWH.VehiclePhysics.Effects.Update ( )
```

6.24.3 Member Data Documentation

6.24.3.1 `exhaustFlash`

```
Backfire NWH.VehiclePhysics.Effects.exhaustFlash = new Backfire()
```

6.24.3.2 exhausts

```
ExhaustSmoke NWH.VehiclePhysics.Effects.exhausts = new ExhaustSmoke()
```

6.24.3.3 lights

```
Lights NWH.VehiclePhysics.Effects.lights = new Lights()
```

6.24.3.4 skidmarks

```
Skidmarks NWH.VehiclePhysics.Effects.skidmarks = new Skidmarks()
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/[Effects.cs](#)

6.25 NWH.VehiclePhysics.Engine Class Reference

Everything related to a vehicle's engine and it's systems.

Public Member Functions

- void [Start](#) ()
Starts the engine. Can be interrupted by calling [Stop\(\)](#). Will not work if there is no fuel or vehicle is damaged over damage threshold.
- void [Stop](#) ()
Stops the engine. Can be interrupted by calling [Start\(\)](#).
- void [Toggle](#) ()
Toggles the engine state using [Start\(\)](#) and [Stop\(\)](#).
- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()

Public Attributes

- bool `runOnStartup` = false
Determines the state of the engine.
- bool `starting` = false
True when engine is starting.
- bool `stopping` = false
True when engine is stopping.
- float `minRPM` = 600
Minimum RPM that engine will run at. RPM can not go below this value.
- float `maxRPM` = 5000
Maximum RPM that engine will run at.
- float `maxPower` = 150
Power at the peak of the power curve.
- float `maxRpmChange` = 10000
Maximum RPM change per second when engine is running without load or when wheels are slipping. Can be used to immitate the flywheel.
- float `throttleSmoothing` = 0.2f
Power delivery smoothing so that the vehicle does not go from 0 to full power instantly. Number represents time needed to reach the input [s]. Smoothing only works on throttle, off throttle there is no smoothing.
- AnimationCurve `powerCurve`
Curve showing how power (Y axis) depends on RPM of the engine (shown on X axis as percentage).
- `ForcedInduction forcedInduction` = new `ForcedInduction()`

Properties

- bool `IsRunning` [get]
Returns true if engine is running. To start or stop the engine call `Start()` or `Stop()` respectively.
- float `StartingPercent` [get]
Value from 0 to 1 determining how far into the starting process the engine is.
- float `StoppingPercent` [get]
Value from 0 to 1 determining how far into the stopping process the engine is.
- bool `FuelCutoff` [get]
State of fuel cutoff.
- float `RPM` [get]
Current RPM of the engine.
- float `RpmOverflow` [get]
0 if engine is operating between minRPM and maxRPM. Larger then 0 when engine RPM is over maxRPM and less than 0 when engine is under minRPM. This is single-frame overflow only, before RPM gets clamped.
- float `RPMPercent` [get]
Value in between 0 and 1 where 0 indicates that engine is at minRPM and 1 indicates that engine is at maxRPM.
- float `Power` [get]
Current power of the engine derived from the engine RPM and the powerCurve.
- float `PowerInHP` [get]
Power in horsepower instead of kW.
- float `TcsPowerReduction` [get, set]
Power reduction from TCS kicking in.
- float `TotalPowerReduction` [get]
Power reduction from all the vehicle's systems.
- float `Torque` [get]
Torque engine is putting out.

6.25.1 Detailed Description

Everything related to a vehicle's engine and it's systems.

6.25.2 Member Function Documentation

6.25.2.1 Initialize()

```
void NWH.VehiclePhysics.Engine.Initialize (
    VehicleController vc )
```

6.25.2.2 Start()

```
void NWH.VehiclePhysics.Engine.Start ( )
```

Starts the engine. Can be interrupted by calling [Stop\(\)](#). Will not work if there is no fuel or vehicle is damaged over damage threshold.

6.25.2.3 Stop()

```
void NWH.VehiclePhysics.Engine.Stop ( )
```

Stops the engine. Can be interrupted by calling [Start\(\)](#).

6.25.2.4 Toggle()

```
void NWH.VehiclePhysics.Engine.Toggle ( )
```

Toggles the engine state using [Start\(\)](#) and [Stop\(\)](#).

6.25.2.5 Update()

```
void NWH.VehiclePhysics.Engine.Update ( )
```

6.25.3 Member Data Documentation

6.25.3.1 forcedInduction

```
ForcedInduction NWH.VehiclePhysics.Engine.forcedInduction = new ForcedInduction()
```

6.25.3.2 maxPower

```
float NWH.VehiclePhysics.Engine.maxPower = 150
```

Power at the peak of the power curve.

6.25.3.3 maxRPM

```
float NWH.VehiclePhysics.Engine.maxRPM = 5000
```

Maximum RPM that engine will run at.

6.25.3.4 maxRpmChange

```
float NWH.VehiclePhysics.Engine.maxRpmChange = 10000
```

Maximum RPM change per second when engine is running without load or when wheels are slipping. Can be used to immitate the flywheel.

6.25.3.5 minRPM

```
float NWH.VehiclePhysics.Engine.minRPM = 600
```

Minimum RPM that engine will run at. RPM can not go below this value.

6.25.3.6 powerCurve

AnimationCurve NWH.VehiclePhysics.Engine.powerCurve

Initial value:

```
= new AnimationCurve(new Keyframe[3] {  
    new Keyframe(0f, 0f),  
    new Keyframe(0.75f, 1f),  
    new Keyframe(1f, 0.92f)  
})
```

Curve showing how power (Y axis) depends on RPM of the engine (shown on X axis as percentage).

6.25.3.7 runOnStartup

bool NWH.VehiclePhysics.Engine.runOnStartup = false

Determines the state of the engine.

6.25.3.8 starting

bool NWH.VehiclePhysics.Engine.starting = false

True when engine is starting.

6.25.3.9 stopping

bool NWH.VehiclePhysics.Engine.stopping = false

True when engine is stopping.

6.25.3.10 throttleSmoothing

float NWH.VehiclePhysics.Engine.throttleSmoothing = 0.2f

Power delivery smoothing so that the vehicle does not go from 0 to full power instantly. Number represents time needed to reach the input [s]. Smoothing only works on throttle, off throttle there is no smoothing.

6.25.4 Property Documentation

6.25.4.1 FuelCutoff

```
bool NWH.VehiclePhysics.Engine.FuelCutoff [get]
```

State of fuel cutoff.

6.25.4.2 IsRunning

```
bool NWH.VehiclePhysics.Engine.IsRunning [get]
```

Returns true if engine is running. To start or stop the engine call [Start\(\)](#) or [Stop\(\)](#) respectively.

6.25.4.3 Power

```
float NWH.VehiclePhysics.Engine.Power [get]
```

Current power of the engine derived from the engine RPM and the powerCurve.

6.25.4.4 PowerInHP

```
float NWH.VehiclePhysics.Engine.PowerInHP [get]
```

Power in horsepower instead of kW.

6.25.4.5 RPM

```
float NWH.VehiclePhysics.Engine.RPM [get]
```

Current RPM of the engine.

6.25.4.6 RpmOverflow

```
float NWH.VehiclePhysics.Engine.RpmOverflow [get]
```

0 if engine is operating between minRPM and maxRPM. Larger then 0 when engine RPM is over maxRPM and less than 0 when engine is under minRPM. This is single-frame overflow only, before RPM gets clamped.

6.25.4.7 RPMPercent

```
float NWH.VehiclePhysics.Engine.RPMPercent [get]
```

Value in between 0 and 1 where 0 indicates that engine is at minRPM and 1 indicates that engine is at maxRPM.

6.25.4.8 StartingPercent

```
float NWH.VehiclePhysics.Engine.StartingPercent [get]
```

Value from 0 to 1 determining how far into the starting process the engine is.

6.25.4.9 StoppingPercent

```
float NWH.VehiclePhysics.Engine.StoppingPercent [get]
```

Value from 0 to 1 determining how far into the stopping process the engine is.

6.25.4.10 TcsPowerReduction

```
float NWH.VehiclePhysics.Engine.TcsPowerReduction [get], [set]
```

Power reduction from TCS kicking in.

6.25.4.11 Torque

```
float NWH.VehiclePhysics.Engine.Torque [get]
```

Torque engine is putting out.

6.25.4.12 TotalPowerReduction

```
float NWH.VehiclePhysics.Engine.TotalPowerReduction [get]
```

Power reduction from all the vehicle's systems.

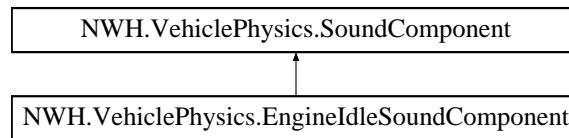
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Engine.cs](#)

6.26 NWH.VehiclePhysics.EngineIdleSoundComponent Class Reference

Sound of an engine idling.

Inheritance diagram for NWH.VehiclePhysics.EngineIdleSoundComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, [AudioMixerGroup](#) amg)
- override void [Update](#) ()

Public Attributes

- float [volumeRange](#) = 0.5f
Volume added to the base engine volume depending on engine state.
- float [pitchRange](#) = 1.5f
Pitch added to the base engine pitch depending on engine RPM.
- float [smoothing](#) = 0.1f
Smoothing of engine sound.
- float [maxDistortion](#) = 0.4f
Distortion that will be added to the engine sound through mixer when under heavy load / high RPM.

Additional Inherited Members

6.26.1 Detailed Description

Sound of an engine idling.

6.26.2 Member Function Documentation

6.26.2.1 Initialize()

```

override void NWH.VehiclePhysics.EngineIdleSoundComponent.Initialize (
    VehicleController vc,
    AudioMixerGroup amg ) [virtual]
  
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.26.2.2 Update()

```
override void NWH.VehiclePhysics.EngineIdleSoundComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.26.3 Member Data Documentation

6.26.3.1 maxDistortion

```
float NWH.VehiclePhysics.EngineIdleSoundComponent.maxDistortion = 0.4f
```

Distortion that will be added to the engine sound through mixer when under heavy load / high RPM.

6.26.3.2 pitchRange

```
float NWH.VehiclePhysics.EngineIdleSoundComponent.pitchRange = 1.5f
```

Pitch added to the base engine pitch depending on engine RPM.

6.26.3.3 smoothing

```
float NWH.VehiclePhysics.EngineIdleSoundComponent.smoothing = 0.1f
```

Smoothing of engine sound.

6.26.3.4 volumeRange

```
float NWH.VehiclePhysics.EngineIdleSoundComponent.volumeRange = 0.5f
```

Volume added to the base engine volume depending on engine state.

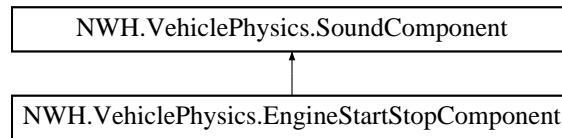
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[EngineIdleComponent.cs](#)

6.27 NWH.VehiclePhysics.EngineStartStopComponent Class Reference

[Sound](#) of an engine starting / stopping. First audio clip is for engine starting, and second one is for engine stopping.

Inheritance diagram for NWH.VehiclePhysics.EngineStartStopComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, AudioManagerGroup amg)
- override void [Update](#) ()

Additional Inherited Members

6.27.1 Detailed Description

[Sound](#) of an engine starting / stopping. First audio clip is for engine starting, and second one is for engine stopping.

6.27.2 Member Function Documentation

6.27.2.1 Initialize()

```
override void NWH.VehiclePhysics.EngineStartStopComponent.Initialize (  
    VehicleController vc,  
    AudioManagerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.27.2.2 Update()

```
override void NWH.VehiclePhysics.EngineStartStopComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[EngineStartStopComponent.cs](#)

6.28 NWH.VehiclePhysics.ExhaustSmoke Class Reference

Controls particle emitters that represent exhausts.

Public Member Functions

- void `Initialize` (`VehicleController` vc)
- void `Update` ()

Public Attributes

- bool `emit` = false
Particles will emit from the exhaust if set to true.
- float `baseIntensity` = 12f
Exhaust smoke intensity when vehicle is idle.
- float `intensityRange` = 20f
Intensity range which will be added to the base intensity depending on engine state.
- float `soot` = 0.4f
Amount of soot that will be present in the final color when engine is under heavy load.
- float `startSize` = 0.3f
Size of each particle at start.
- float `lifeDistance` = 2.5f
How far behind the vehicle will exhaust particles extend?
- Color `vaporColor` = new Color(0.7f, 0.7f, 0.7f, 0.35f)
Color of exhaust at idle.
- Color `sootColor` = new Color(0.1f, 0.1f, 0.1f, 0.5f)
Color of exhaust at heavy load.
- List< ParticleSystem > `particleSystems` = new List<ParticleSystem>()
List of particle systems representing exhausts.

6.28.1 Detailed Description

Controls particle emitters that represent exhausts.

6.28.2 Member Function Documentation

6.28.2.1 Initialize()

```
void NWH.VehiclePhysics.ExhaustSmoke.Initialize (  
    VehicleController vc )
```

6.28.2.2 Update()

```
void NWH.VehiclePhysics.ExhaustSmoke.Update ( )
```

6.28.3 Member Data Documentation

6.28.3.1 baseIntensity

```
float NWH.VehiclePhysics.ExhaustSmoke.baseIntensity = 12f
```

Exhaust smoke intensity when vehicle is idle.

6.28.3.2 emit

```
bool NWH.VehiclePhysics.ExhaustSmoke.emit = false
```

Particles will emit from the exhaust if set to true.

6.28.3.3 intensityRange

```
float NWH.VehiclePhysics.ExhaustSmoke.intensityRange = 20f
```

Intensity range which will be added to the base intensity depending on engine state.

6.28.3.4 lifeDistance

```
float NWH.VehiclePhysics.ExhaustSmoke.lifeDistance = 2.5f
```

How far behind the vehicle will exhaust particles extend?

6.28.3.5 particleSystems

```
List<ParticleSystem> NWH.VehiclePhysics.ExhaustSmoke.particleSystems = new List<Particle↵  
System> ()
```

List of particle systems representing exhausts.

6.28.3.6 soot

```
float NWH.VehiclePhysics.ExhaustSmoke.soot = 0.4f
```

Amount of soot that will be present in the final color when engine is under heavy load.

6.28.3.7 sootColor

```
Color NWH.VehiclePhysics.ExhaustSmoke.sootColor = new Color(0.1f, 0.1f, 0.1f, 0.5f)
```

Color of exhaust at heavy load.

6.28.3.8 startSize

```
float NWH.VehiclePhysics.ExhaustSmoke.startSize = 0.3f
```

Size of each particle at start.

6.28.3.9 vaporColor

```
Color NWH.VehiclePhysics.ExhaustSmoke.vaporColor = new Color(0.7f, 0.7f, 0.7f, 0.35f)
```

Color of exhaust at idle.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/[ExhaustSmoke.cs](#)

6.29 NWH.VehiclePhysics.FlipOver Class Reference

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()

Public Attributes

- bool `enabled` = true
Should the vehicle be rotated back when it flips over?
- float `timeout` = 3f
Time after detecting flip over after which vehicle will be flipped back.
- float `allowedAngle` = 70f
Minimum angle that the vehicle needs to be at for it to be detected as flipped over.
- float `maxDetectionSpeed` = 1f
Flip over detection will be disabled if velocity is above this value [m/s].
- float `rotationSpeed` = 80f
Rotation speed of the vehicle while being flipped back.
- bool `flippedOver` = false
Is the vehicle flipped over?

6.29.1 Member Function Documentation

6.29.1.1 Initialize()

```
void NWH.VehiclePhysics.FlipOver.Initialize (
    VehicleController vc )
```

6.29.1.2 Update()

```
void NWH.VehiclePhysics.FlipOver.Update ( )
```

6.29.2 Member Data Documentation

6.29.2.1 allowedAngle

```
float NWH.VehiclePhysics.FlipOver.allowedAngle = 70f
```

Minimum angle that the vehicle needs to be at for it to be detected as flipped over.

6.29.2.2 enabled

```
bool NWH.VehiclePhysics.FlipOver.enabled = true
```

Should the vehicle be rotated back when it flips over?

6.29.2.3 flippedOver

```
bool NWH.VehiclePhysics.FlipOver.flippedOver = false
```

Is the vehicle flipped over?

6.29.2.4 maxDetectionSpeed

```
float NWH.VehiclePhysics.FlipOver.maxDetectionSpeed = 1f
```

Flip over detection will be disabled if velocity is above this value [m/s].

6.29.2.5 rotationSpeed

```
float NWH.VehiclePhysics.FlipOver.rotationSpeed = 80f
```

Rotation speed of the vehicle while being flipped back.

6.29.2.6 timeout

```
float NWH.VehiclePhysics.FlipOver.timeout = 3f
```

Time after detecting flip over after which vehicle will be flipped back.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[FlipOver.cs](#)

6.30 NWH.VehiclePhysics.ForcedInduction Class Reference

Supercharger, turbocharger, etc. Can also be used on vehicles with no forced induction for sound effects such as intake noise or engine fan noise.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()

Public Attributes

- bool `useForcedInduction` = true
Should forced induction be used?
- float `spoolUpTime` = 1f
Shortest time possible needed for turbo to spool up to its maximum RPM.
- float `maxPowerGainMultiplier` = 0.4f
Additional power that will be added to the engine's power. This is the maximum value possible and depends on spool percent.
- bool `flutterSoundFlag` = false
Flutter can sound when true.

Properties

- float `SpoolPercent` [get]
Percent of forced induction's RPM in relation to its max RPM.
- float `PowerGainMultiplier` [get]
Current power gained from forced induction.

6.30.1 Detailed Description

Supercharger, turbocharger, etc. Can also be used on vehicles with no forced induction for sound effects such as intake noise or engine fan noise.

6.30.2 Member Function Documentation

6.30.2.1 Initialize()

```
void NWH.VehiclePhysics.ForcedInduction.Initialize (  
    VehicleController vc )
```

6.30.2.2 Update()

```
void NWH.VehiclePhysics.ForcedInduction.Update ( )
```

6.30.3 Member Data Documentation

6.30.3.1 flutterSoundFlag

```
bool NWH.VehiclePhysics.ForcedInduction.flutterSoundFlag = false
```

Flutter can sound when true.

6.30.3.2 maxPowerGainMultiplier

```
float NWH.VehiclePhysics.ForcedInduction.maxPowerGainMultiplier = 0.4f
```

Additional power that will be added to the engine's power. This is the maximum value possible and depends on spool percent.

6.30.3.3 spoolUpTime

```
float NWH.VehiclePhysics.ForcedInduction.spoolUpTime = 1f
```

Shortest time possible needed for turbo to spool up to its maximum RPM.

6.30.3.4 useForcedInduction

```
bool NWH.VehiclePhysics.ForcedInduction.useForcedInduction = true
```

Should forced induction be used?

6.30.4 Property Documentation

6.30.4.1 PowerGainMultiplier

```
float NWH.VehiclePhysics.ForcedInduction.PowerGainMultiplier [get]
```

Current power gained from forced induction.

6.30.4.2 SpoolPercent

```
float NWH.VehiclePhysics.ForcedInduction.SpoolPercent [get]
```

Percent of forced induction's RPM in relation to its max RPM.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[ForcedInduction.cs](#)

6.31 NWH.WheelController3D.WheelController.Friction Class Reference

All info related to longitudinal force calculation.

Public Attributes

- float [forceCoefficient](#) = 1.1f
- float [slipCoefficient](#) = 1
- float [maxForce](#)
- float [slip](#)
- float [speed](#)
- float [force](#)

6.31.1 Detailed Description

All info related to longitudinal force calculation.

6.31.2 Member Data Documentation

6.31.2.1 force

```
float NWH.WheelController3D.WheelController.Friction.force
```

6.31.2.2 forceCoefficient

```
float NWH.WheelController3D.WheelController.Friction.forceCoefficient = 1.1f
```

6.31.2.3 maxForce

```
float NWH.WheelController3D.WheelController.Friction.maxForce
```

6.31.2.4 slip

```
float NWH.WheelController3D.WheelController.Friction.slip
```

6.31.2.5 slipCoefficient

```
float NWH.WheelController3D.WheelController.Friction.slipCoefficient = 1
```

6.31.2.6 speed

```
float NWH.WheelController3D.WheelController.Friction.speed
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/[WheelController.cs](#)

6.32 NWH.WheelController3D.WheelController.FrictionPreset Class Reference

Container class for holding wheel friction presets.

Public Types

- enum [FrictionPresetEnum](#) {
[FrictionPresetEnum.TarmacDry](#), [FrictionPresetEnum.TarmacWet](#), [FrictionPresetEnum.Gravel](#), [FrictionPresetEnum.Grass](#),
[FrictionPresetEnum.Sand](#), [FrictionPresetEnum.Snow](#), [FrictionPresetEnum.Ice](#), [FrictionPresetEnum.Generic](#),
[FrictionPresetEnum.Tracks](#), [FrictionPresetEnum.Arcade](#) }

Public Member Functions

- [FrictionPreset](#) (string [name](#), Vector4 [BCDE](#))

Static Public Member Functions

- static AnimationCurve [GenerateFrictionCurve](#) (Vector4 p)
Generate Curve from B,C,D and E parameters of Pacejka's simplified magic formula

Public Attributes

- string [name](#)
- Vector4 [BCDE](#)

Static Public Attributes

- static [FrictionPreset TarmacDry](#) = new [FrictionPreset](#)("TarmacDry", new Vector4(12.5f, 2.05f, 0.92f, 0.97f))
- static [FrictionPreset TarmacWet](#) = new [FrictionPreset](#)("TarmacWet", new Vector4(13.2f, 2.35f, 0.82f, 1.0f))
- static [FrictionPreset Gravel](#) = new [FrictionPreset](#)("Gravel", new Vector4(9.0f, 1.1f, 0.8f, 1f))
- static [FrictionPreset Grass](#) = new [FrictionPreset](#)("Grass", new Vector4(8.4f, 1.3f, 0.5f, 0.4f))
- static [FrictionPreset Sand](#) = new [FrictionPreset](#)("Sand", new Vector4(8.0f, 1.2f, 0.6f, 0.5f))
- static [FrictionPreset Snow](#) = new [FrictionPreset](#)("Snow", new Vector4(8.5f, 1.1f, 0.4f, 0.9f))
- static [FrictionPreset Ice](#) = new [FrictionPreset](#)("Ice", new Vector4(4.0f, 2.0f, 0.1f, 1.0f))
- static [FrictionPreset Generic](#) = new [FrictionPreset](#)("Generic", new Vector4(8.0f, 1.9f, 0.8f, 0.99f))
- static [FrictionPreset Tracks](#) = new [FrictionPreset](#)("Tracks", new Vector4(0.1f, 2f, 15f, 1f))
- static [FrictionPreset Arcade](#) = new [FrictionPreset](#)("Arcade", new Vector4(4f, 1f, 2f, 0.5f))
- static List< [FrictionPreset](#) > [FrictionPresetList](#)

Properties

- AnimationCurve [Curve](#) [get]

6.32.1 Detailed Description

Container class for holding wheel friction presets.

6.32.2 Member Enumeration Documentation

6.32.2.1 FrictionPresetEnum

```
enum NWH.WheelController3D.WheelController.FrictionPreset.FrictionPresetEnum [strong]
```

Enumerator

TarmacDry	
TarmacWet	
Gravel	
Grass	
Sand	
Snow	
Ice	
Generic	
Tracks	
Arcade	

6.32.3 Constructor & Destructor Documentation

6.32.3.1 FrictionPreset()

```
NWH.WheelController3D.WheelController.FrictionPreset.FrictionPreset (
    string name,
    Vector4 BCDE )
```

6.32.4 Member Function Documentation

6.32.4.1 GenerateFrictionCurve()

```
static AnimationCurve NWH.WheelController3D.WheelController.FrictionPreset.GenerateFriction↵
Curve (
    Vector4 p ) [static]
```

Generate Curve from B,C,D and E parameters of Pacejka's simplified magic formula

6.32.5 Member Data Documentation

6.32.5.1 Arcade

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.Arcade = new FrictionPreset ("Arcade",
new Vector4(4f, 1f, 2f, 0.5f)) [static]
```

6.32.5.2 BCDE

```
Vector4 NWH.WheelController3D.WheelController.FrictionPreset.BCDE
```

6.32.5.3 FrictionPresetList

```
List<FrictionPreset> NWH.WheelController3D.WheelController.FrictionPreset.FrictionPresetList
[static]
```

6.32.5.4 Generic

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.Generic = new FrictionPreset("Generic",  
new Vector4(8.0f, 1.9f, 0.8f, 0.99f)) [static]
```

6.32.5.5 Grass

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.Grass = new FrictionPreset("Grass",  
new Vector4(8.4f, 1.3f, 0.5f, 0.4f)) [static]
```

6.32.5.6 Gravel

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.Gravel = new FrictionPreset("Gravel",  
new Vector4(9.0f, 1.1f, 0.8f, 1f)) [static]
```

6.32.5.7 Ice

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.Ice = new FrictionPreset("Ice",  
new Vector4(4.0f, 2.0f, 0.1f, 1.0f)) [static]
```

6.32.5.8 name

```
string NWH.WheelController3D.WheelController.FrictionPreset.name
```

6.32.5.9 Sand

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.Sand = new FrictionPreset("Sand",  
new Vector4(8.0f, 1.2f, 0.6f, 0.5f)) [static]
```

6.32.5.10 Snow

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.Snow = new FrictionPreset("Snow",  
new Vector4(8.5f, 1.1f, 0.4f, 0.9f)) [static]
```

6.32.5.11 TarmacDry

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.TarmacDry = new FrictionPreset("TarmacDry", new Vector4(12.5f, 2.05f, 0.92f, 0.97f)) [static]
```

6.32.5.12 TarmacWet

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.TarmacWet = new FrictionPreset("TarmacWet", new Vector4(13.2f, 2.35f, 0.82f, 1.0f)) [static]
```

6.32.5.13 Tracks

```
FrictionPreset NWH.WheelController3D.WheelController.FrictionPreset.Tracks = new FrictionPreset("Tracks", new Vector4(0.1f, 2f, 15f, 1f)) [static]
```

6.32.6 Property Documentation

6.32.6.1 Curve

```
AnimationCurve NWH.WheelController3D.WheelController.FrictionPreset.Curve [get]
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/[WheelController.FrictionPreset.cs](#)

6.33 NWH.VehiclePhysics.Fuel Class Reference

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()

Public Attributes

- bool [useFuel](#) = false
Should fuel be used? If set to false HasFuel will always return true.
- float [capacity](#) = 50f
Fuel capacity in liters.
- float [amount](#) = 50f
Maximum amount in liters that the fuel tank can hold.
- float [efficiency](#) = 0.45f
Engine efficiency (in percent). 1 would mean that all the energy contained in fuel would go into output power.

Properties

- bool [HasFuel](#) [get]
True if has fuel or if use fuel is false.
- float [FuelPercentage](#) [get]
Percentage of fuel from the max amount the tank can hold.
- float [ConsumptionLitersPerSecond](#) [get]
Fuel consumption in liters per second.
- float [ConsumptionMPG](#) [get]
Fuel consumption in miles per gallon.
- float [ConsumptionLitersPer100Kilometers](#) [get]
Fuel consumption in liters per 100 kilometers.
- float [ConsumptionKilometersPerLiter](#) [get]
Fuel consumption in kilometers per liter.

6.33.1 Member Function Documentation

6.33.1.1 Initialize()

```
void NWH.VehiclePhysics.Fuel.Initialize (
    VehicleController vc )
```

6.33.1.2 Update()

```
void NWH.VehiclePhysics.Fuel.Update ( )
```

6.33.2 Member Data Documentation

6.33.2.1 amount

```
float NWH.VehiclePhysics.Fuel.amount = 50f
```

Maximum amount in liters that the fuel tank can hold.

6.33.2.2 capacity

```
float NWH.VehiclePhysics.Fuel.capacity = 50f
```

[Fuel](#) capacity in liters.

6.33.2.3 efficiency

```
float NWH.VehiclePhysics.Fuel.efficiency = 0.45f
```

[Engine](#) efficiency (in percent). 1 would mean that all the energy contained in fuel would go into output power.

6.33.2.4 useFuel

```
bool NWH.VehiclePhysics.Fuel.useFuel = false
```

Should fuel be used? If set to false HasFuel will always return true.

6.33.3 Property Documentation

6.33.3.1 ConsumptionKilometersPerLiter

```
float NWH.VehiclePhysics.Fuel.ConsumptionKilometersPerLiter [get]
```

[Fuel](#) consumption in kilometers per liter.

6.33.3.2 ConsumptionLitersPer100Kilometers

```
float NWH.VehiclePhysics.Fuel.ConsumptionLitersPer100Kilometers [get]
```

[Fuel](#) consumption in liters per 100 kilometers.

6.33.3.3 ConsumptionLitersPerSecond

```
float NWH.VehiclePhysics.Fuel.ConsumptionLitersPerSecond [get]
```

[Fuel](#) consumption in liters per second.

6.33.3.4 ConsumptionMPG

```
float NWH.VehiclePhysics.Fuel.ConsumptionMPG [get]
```

[Fuel](#) consumption in miles per gallon.

6.33.3.5 FuelPercentage

```
float NWH.VehiclePhysics.Fuel.FuelPercentage [get]
```

Percentage of fuel from the max amount the tank can hold.

6.33.3.6 HasFuel

```
bool NWH.VehiclePhysics.Fuel.HasFuel [get]
```

True if has fuel or if use fuel is false.

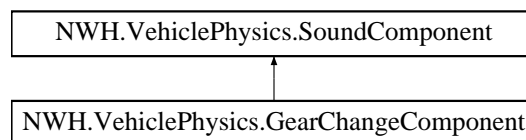
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Fuel.cs](#)

6.34 NWH.VehiclePhysics.GearChangeComponent Class Reference

Shifter sound played when changing gears. Supports multiple audio clips of which one is chosen at random each time this effect is played.

Inheritance diagram for NWH.VehiclePhysics.GearChangeComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController vc](#), [AudioMixerGroup amg](#))
- override void [Update](#) ()

Additional Inherited Members

6.34.1 Detailed Description

Shifter sound played when changing gears. Supports multiple audio clips of which one is chosen at random each time this effect is played.

6.34.2 Member Function Documentation

6.34.2.1 Initialize()

```
override void NWH.VehiclePhysics.GearChangeComponent.Initialize (
    VehicleController vc,
    AudioMixerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.34.2.2 Update()

```
override void NWH.VehiclePhysics.GearChangeComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[GearChangeComponent.cs](#)

6.35 NWH.VehiclePhysics.Axle.Geometry Class Reference

Class holding all geometry related data for axle and it's wheels.

Public Attributes

- float [steerCoefficient](#)
- float [ackermannPercent](#) = 0.15f
- float [toeAngle](#) = 0
- float [casterAngle](#) = 0
- float [camberAtTop](#) = 0
- float [camberAtBottom](#) = 0
- bool [isSolid](#) = false
- float [antiRollBarForce](#)

6.35.1 Detailed Description

Class holding all geometry related data for axle and it's wheels.

6.35.2 Member Data Documentation

6.35.2.1 ackermannPercent

```
float NWH.VehiclePhysics.Axle.Geometry.ackermannPercent = 0.15f
```

6.35.2.2 antiRollBarForce

```
float NWH.VehiclePhysics.Axle.Geometry.antiRollBarForce
```

6.35.2.3 camberAtBottom

```
float NWH.VehiclePhysics.Axle.Geometry.camberAtBottom = 0
```

6.35.2.4 camberAtTop

```
float NWH.VehiclePhysics.Axle.Geometry.camberAtTop = 0
```

6.35.2.5 casterAngle

```
float NWH.VehiclePhysics.Axle.Geometry.casterAngle = 0
```

6.35.2.6 isSolid

```
bool NWH.VehiclePhysics.Axle.Geometry.isSolid = false
```

6.35.2.7 steerCoefficient

```
float NWH.VehiclePhysics.Axle.Geometry.steerCoefficient
```

6.35.2.8 toeAngle

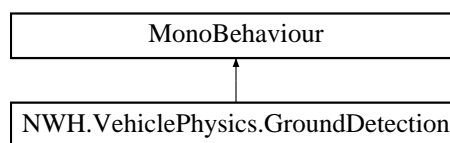
```
float NWH.VehiclePhysics.Axle.Geometry.toeAngle = 0
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Axle.cs](#)

6.36 NWH.VehiclePhysics.GroundDetection Class Reference

Inheritance diagram for NWH.VehiclePhysics.GroundDetection:



Classes

- class [GroundEntity](#)
A class representing a single ground surface type.
- class [SoundComponent](#)
Base class for surface sounds.

Public Member Functions

- int [GetCurrentGroundEntityIndex](#) ([WheelController](#) wheelController)
Gets index of current ground entity in ground entities list.
- Material [GetCurrentSkidmarkTexture](#) ([WheelController](#) wheelController)
Returns skidmark material for the current texture.
- [GroundEntity](#) [GetCurrentGroundEntity](#) ([WheelController](#) wheelController)
Gets the ground entity the wheel is currently on.

Public Attributes

- GameObject [smokePrefab](#)
Prefab of the particle system for generating smoke as a result of spinning or slipping tires.
- GameObject [dustPrefab](#)
Prefab of the particle system for generating dust as a result of traveling over sand, gravel, etc.
- List< [GroundEntity](#) > [groundEntities](#) = new List<[GroundEntity](#)>()
Ground entities - each represents a single ground surface.

6.36.1 Member Function Documentation

6.36.1.1 GetCurrentGroundEntity()

```
GroundEntity NWH.VehiclePhysics.GroundDetection.GetCurrentGroundEntity (
    WheelController wheelController )
```

Gets the ground entity the wheel is currently on.

6.36.1.2 GetCurrentGroundEntityIndex()

```
int NWH.VehiclePhysics.GroundDetection.GetCurrentGroundEntityIndex (
    WheelController wheelController )
```

Gets index of current ground entity in ground entities list.

6.36.1.3 GetCurrentSkidmarkTexture()

```
Material NWH.VehiclePhysics.GroundDetection.GetCurrentSkidmarkTexture (
    WheelController wheelController )
```

Returns skidmark material for the current texture.

6.36.2 Member Data Documentation

6.36.2.1 dustPrefab

```
GameObject NWH.VehiclePhysics.GroundDetection.dustPrefab
```

Prefab of the particle system for generating dust as a result of traveling over sand, gravel, etc.

6.36.2.2 groundEntities

```
List<GroundEntity> NWH.VehiclePhysics.GroundDetection.groundEntities = new List<GroundEntity>()
```

Ground entities - each represents a single ground surface.

6.36.2.3 smokePrefab

`GameObject NWH.VehiclePhysics.GroundDetection.smokePrefab`

Prefab of the particle system for generating smoke as a result of spinning or slipping tires.

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

- `E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/`[GroundDetection.cs](#)

6.37 NWH.VehiclePhysics.GroundDetection.GroundEntity Class Reference

A class representing a single ground surface type.

Public Attributes

- string [name](#)
Name of the ground entity.
- `List< int >` [terrainTextureIndices](#) = `new List<int>()`
Indices of terrain textures that represent this type of surface. Starts with 0 with the first texture being in the top left corner under terrain settings - Paint Texture.
- `List< string >` [tags](#) = `new List<string>()`
Objects with tags in this list will be recognized as this type of surface.
- [WheelController.FrictionPreset.FrictionPresetEnum](#) [frictionPresetEnum](#)
Friction preset of WC3D that will be used for this surface. More presets can be added in [WheelController.FrictionPresets](#).
- Material [skidmarkMaterial](#)
Material that will be used when generating skidmarks.
- bool [slipBasedSkidIntensity](#) = `false`
If set to true sidmarks will be dependent on wheel's slip (true for asphalt, concrete, etc.).
- float [smokeIntensity](#) = `30f`
Intensity of smoke generated by wheel slip.
- float [dustIntensity](#) = `0f`
Intensity of dust generated by wheel traveling over sand, gravel, etc.
- Color [dustColor](#) = `Color.yellow`
Color of dust generated by wheel traveling over sand, gravel, etc.
- bool [slipSensitiveSurfaceSound](#)
If set to true surface volume will be dependent on slip (asphalt, concrete, etc.). Set to false for dirt, grass and other soft surfaces.
- [SoundComponent](#) [surfaceSoundComponent](#)
[Sound](#) generated by tire rolling over surface.
- [SoundComponent](#) [skidSoundComponent](#)
[Sound](#) produced by tire skidding over surface.

6.37.1 Detailed Description

A class representing a single ground surface type.

6.37.2 Member Data Documentation

6.37.2.1 dustColor

```
Color NWH.VehiclePhysics.GroundDetection.GroundEntity.dustColor = Color.yellow
```

Color of dust generated by wheel traveling over sand, gravel, etc.

6.37.2.2 dustIntensity

```
float NWH.VehiclePhysics.GroundDetection.GroundEntity.dustIntensity = 0f
```

Intensity of dust generated by wheel traveling over sand, gravel, etc.

6.37.2.3 frictionPresetEnum

```
WheelController.FrictionPreset.FrictionPresetEnum NWH.VehiclePhysics.GroundDetection.GroundEntity.frictionPresetEnum
```

Friction preset of WC3D that will be used for this surface. More presets can be added in WheelController.FrictionPresets.

6.37.2.4 name

```
string NWH.VehiclePhysics.GroundDetection.GroundEntity.name
```

Name of the ground entity.

6.37.2.5 skidmarkMaterial

```
Material NWH.VehiclePhysics.GroundDetection.GroundEntity.skidmarkMaterial
```

Material that will be used when generating skidmarks.

6.37.2.6 skidSoundComponent

[SoundComponent](#) NWH.VehiclePhysics.GroundDetection.GroundEntity.skidSoundComponent

[Sound](#) produced by tire skidding over surface.

6.37.2.7 slipBasedSkidIntensity

```
bool NWH.VehiclePhysics.GroundDetection.GroundEntity.slipBasedSkidIntensity = false
```

If set to true sidmarks will be dependent on wheel's slip (true for asphalt, concrete, etc.).

6.37.2.8 slipSensitiveSurfaceSound

```
bool NWH.VehiclePhysics.GroundDetection.GroundEntity.slipSensitiveSurfaceSound
```

If set to true surface volume will be dependent on slip (asphalt, concrete, etc.). Set to false for dirt, grass and other soft surfaces.

6.37.2.9 smokeIntensity

```
float NWH.VehiclePhysics.GroundDetection.GroundEntity.smokeIntensity = 30f
```

Intensity of smoke generated by wheel slip.

6.37.2.10 surfaceSoundComponent

[SoundComponent](#) NWH.VehiclePhysics.GroundDetection.GroundEntity.surfaceSoundComponent

[Sound](#) generated by tire rolling over surface.

6.37.2.11 tags

```
List<string> NWH.VehiclePhysics.GroundDetection.GroundEntity.tags = new List<string>()
```

Objects with tags in this list will be recognized as this type of surface.

6.37.2.12 terrainTextureIndices

```
List<int> NWH.VehiclePhysics.GroundDetection.GroundEntity.terrainTextureIndices = new List<int>()
```

Indices of terrain textures that represent this type of surface. Starts with 0 with the first texture being in the top left corner under terrain settings - Paint Texture.

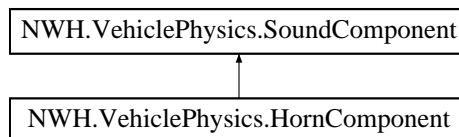
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[GroundDetection.cs](#)

6.38 NWH.VehiclePhysics.HornComponent Class Reference

Vehicle horn sound.

Inheritance diagram for NWH.VehiclePhysics.HornComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, AudioManagerGroup amg)
- override void [Update](#) ()

Additional Inherited Members

6.38.1 Detailed Description

Vehicle horn sound.

6.38.2 Member Function Documentation

6.38.2.1 Initialize()

```
override void NWH.VehiclePhysics.HornComponent.Initialize (
    VehicleController vc,
    AudioManagerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.38.2.2 Update()

```
override void NWH.VehiclePhysics.HornComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[HornComponent.cs](#)

6.39 NWH.VehiclePhysics.InputStates Class Reference

Class for storing input states of the vehicle.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)

Public Attributes

- bool [leftBlinker](#)
- bool [rightBlinker](#)
- bool [lowBeamLights](#)
- bool [fullBeamLights](#)
- bool [hazardLights](#)
- bool [trailerAttachDetach](#)

Trailer will be attached only if under the threshold set in trailer options.

- bool [horn](#)

Properties

- bool [ShiftUp](#) [get, set]
- bool [ShiftDown](#) [get, set]
- float [Horizontal](#) [get, set]
- float [RawVertical](#) [get]

Returns vertical input without any processing.

- float [Vertical](#) [get, set]
- float [Clutch](#) [get, set]
- float [Handbrake](#) [get, set]

6.39.1 Detailed Description

Class for storing input states of the vehicle.

6.39.2 Member Function Documentation

6.39.2.1 Initialize()

```
void NWH.VehiclePhysics.InputStates.Initialize (
    VehicleController vc )
```

6.39.3 Member Data Documentation

6.39.3.1 fullBeamLights

```
bool NWH.VehiclePhysics.InputStates.fullBeamLights
```

6.39.3.2 hazardLights

```
bool NWH.VehiclePhysics.InputStates.hazardLights
```

6.39.3.3 horn

```
bool NWH.VehiclePhysics.InputStates.horn
```

6.39.3.4 leftBlinker

```
bool NWH.VehiclePhysics.InputStates.leftBlinker
```

6.39.3.5 lowBeamLights

```
bool NWH.VehiclePhysics.InputStates.lowBeamLights
```

6.39.3.6 rightBlinker

```
bool NWH.VehiclePhysics.InputStates.rightBlinker
```

6.39.3.7 trailerAttachDetach

```
bool NWH.VehiclePhysics.InputStates.trailerAttachDetach
```

Trailer will be attached only if under the threshold set in trailer options.

6.39.4 Property Documentation

6.39.4.1 Clutch

```
float NWH.VehiclePhysics.InputStates.Clutch [get], [set]
```

6.39.4.2 Handbrake

```
float NWH.VehiclePhysics.InputStates.Handbrake [get], [set]
```

6.39.4.3 Horizontal

```
float NWH.VehiclePhysics.InputStates.Horizontal [get], [set]
```

6.39.4.4 RawVertical

```
float NWH.VehiclePhysics.InputStates.RawVertical [get]
```

Returns vertical input without any processing.

6.39.4.5 ShiftDown

```
bool NWH.VehiclePhysics.InputStates.ShiftDown [get], [set]
```

6.39.4.6 ShiftUp

```
bool NWH.VehiclePhysics.InputStates.ShiftUp [get], [set]
```

6.39.4.7 Vertical

```
float NWH.VehiclePhysics.InputStates.Vertical [get], [set]
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/[InputStates.cs](#)

6.40 NWH.VehiclePhysics.Lights Class Reference

Class for controlling all of the vehicle lights.

Classes

- class [VehicleLight](#)
Single vehicle light.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [TurnOffAllLights](#) ()
Turns off all lights and emission on all meshes.
- void [Update](#) ()

Public Attributes

- bool [enabled](#) = true
Determines the state of all lights.
- [VehicleLight](#) [stopLights](#) = new [VehicleLight](#)()
Rear lights that will light up when brake is pressed.
- [VehicleLight](#) [rearLights](#) = new [VehicleLight](#)()
Rear [Lights](#) that will light up when headlights are on.
- [VehicleLight](#) [headLights](#) = new [VehicleLight](#)()
Low beam lights.
- [VehicleLight](#) [fullBeams](#) = new [VehicleLight](#)()
High (full) beam lights.
- [VehicleLight](#) [leftBlinkers](#) = new [VehicleLight](#)()
Blinkers on the left side of the vehicle.
- [VehicleLight](#) [rightBlinkers](#) = new [VehicleLight](#)()
Blinkers on the right side of the vehicle.

Properties

- bool `BlinkerState` [get]
State in which blinker is at the moment.

6.40.1 Detailed Description

Class for controlling all of the vehicle lights.

6.40.2 Member Function Documentation

6.40.2.1 Initialize()

```
void NWH.VehiclePhysics.Lights.Initialize (
    VehicleController vc )
```

6.40.2.2 TurnOffAllLights()

```
void NWH.VehiclePhysics.Lights.TurnOffAllLights ( )
```

Turns off all lights and emission on all meshes.

6.40.2.3 Update()

```
void NWH.VehiclePhysics.Lights.Update ( )
```

6.40.3 Member Data Documentation

6.40.3.1 enabled

```
bool NWH.VehiclePhysics.Lights.enabled = true
```

Determines the state of all lights.

6.40.3.2 fullBeams

```
VehicleLight NWH.VehiclePhysics.Lights.fullBeams = new VehicleLight()
```

High (full) beam lights.

6.40.3.3 headLights

```
VehicleLight NWH.VehiclePhysics.Lights.headLights = new VehicleLight()
```

Low beam lights.

6.40.3.4 leftBlinkers

```
VehicleLight NWH.VehiclePhysics.Lights.leftBlinkers = new VehicleLight()
```

Blinkers on the left side of the vehicle.

6.40.3.5 rearLights

```
VehicleLight NWH.VehiclePhysics.Lights.rearLights = new VehicleLight()
```

Rear [Lights](#) that will light up when headlights are on.

6.40.3.6 rightBlinkers

```
VehicleLight NWH.VehiclePhysics.Lights.rightBlinkers = new VehicleLight()
```

Blinkers on the right side of the vehicle.

6.40.3.7 stopLights

```
VehicleLight NWH.VehiclePhysics.Lights.stopLights = new VehicleLight()
```

Rear lights that will light up when brake is pressed.

6.40.4 Property Documentation

6.40.4.1 BlinkerState

```
bool NWH.VehiclePhysics.Lights.BlinkerState [get]
```

State in which blinker is at the moment.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/[Lights.cs](#)

6.41 NWH.VehiclePhysics.Metrics.Metric Class Reference

Public Member Functions

- delegate float [UpdateDelegate](#) ()
- void [Update](#) ([UpdateDelegate](#) del, bool increment)
- void [Reset](#) ()

Public Attributes

- float [value](#) = 0f

6.41.1 Member Function Documentation

6.41.1.1 Reset()

```
void NWH.VehiclePhysics.Metrics.Metric.Reset ( )
```

6.41.1.2 Update()

```
void NWH.VehiclePhysics.Metrics.Metric.Update (
    UpdateDelegate del,
    bool increment )
```


6.41.1.3 UpdateDelegate()

```
delegate float NWH.VehiclePhysics.Metrics.Metric.UpdateDelegate ( )
```

6.41.2 Member Data Documentation

6.41.2.1 value

```
float NWH.VehiclePhysics.Metrics.Metric.value = 0f
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Metrics.cs](#)

6.42 NWH.VehiclePhysics.Metrics Class Reference

Class for holding metrics such as odometer, top speed and drift time. All the speed values are in m/s. If you need the value in km/h or mph use UnitConverter functions.

Classes

- class [Metric](#)

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()

Public Attributes

- [Metric](#) odometer
- [Metric](#) topSpeed
- [Metric](#) averageSpeed
- [Metric](#) totalDriftTime
- [Metric](#) continousDriftTime
- [Metric](#) totalDriftDistance
- [Metric](#) continousDriftDistance

6.42.1 Detailed Description

Class for holding metrics such as odometer, top speed and drift time. All the speed values are in m/s. If you need the value in km/h or mph use UnitConverter functions.

6.42.2 Member Function Documentation

6.42.2.1 Initialize()

```
void NWH.VehiclePhysics.Metrics.Initialize (
    VehicleController vc )
```

6.42.2.2 Update()

```
void NWH.VehiclePhysics.Metrics.Update ( )
```

6.42.3 Member Data Documentation

6.42.3.1 averageSpeed

```
Metric NWH.VehiclePhysics.Metrics.averageSpeed
```

6.42.3.2 continousDriftDistance

```
Metric NWH.VehiclePhysics.Metrics.continuousDriftDistance
```

6.42.3.3 continousDriftTime

```
Metric NWH.VehiclePhysics.Metrics.continuousDriftTime
```

6.42.3.4 odometer

```
Metric NWH.VehiclePhysics.Metrics.odometer
```

6.42.3.5 topSpeed

Metric `NWH.VehiclePhysics.Metrics.topSpeed`

6.42.3.6 totalDriftDistance

Metric `NWH.VehiclePhysics.Metrics.totalDriftDistance`

6.42.3.7 totalDriftTime

Metric `NWH.VehiclePhysics.Metrics.totalDriftTime`

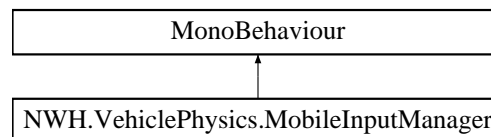
The documentation for this class was generated from the following file:

- `E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Metrics.cs`

6.43 NWH.VehiclePhysics.MobileInputManager Class Reference

Class for handling mobile user input via touch screen and sensors. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.

Inheritance diagram for `NWH.VehiclePhysics.MobileInputManager`:



Public Types

- enum `InputType` { `InputType.Accelerometer`, `InputType.SteeringWheel` }
Steer input device. Accelerometer - uses sensors to get horizontal axis. Screen - uses left side of the screen along with touch input to get steering position. Steering Wheel - uses SteeringWheel script and steering wheel on-screen graphic that can be rotated by dragging.

Public Member Functions

- void `UpdateSteering` ()
- void `ThrottleDown` ()
- void `ThrottleUp` ()
- void `BrakeDown` ()
- void `BrakeUp` ()
- void `ToggleHighBeams` ()
- void `ToggleLowBeams` ()
- void `EngineStartStop` ()
- void `ChangeVehicle` ()
- void `ChangeCamera` ()
- void `TrailerAttachDetach` ()

Public Attributes

- `InputType inputType = InputType.SteeringWheel`
Active steer devices.
- `SteeringWheel steeringWheel`
Steering wheel script. Optional and not needed if `SteeringWheel` option is not used.
- `VehicleChanger vehicleChanger`
Set to null (none) if you want to use your own vehicle controller. If this is set to other than null current active vehicle according to the assigned vehicle changer will be used instead of the assigned vehicle controller.
- `VehicleController vehicleController`
If you want to use this script with a single vehicle or want to set your own vehicle controller from script set vehicle changer field to null / none.
- `float tiltSensitivity = 1.5f`
Higher value will result in higher steer angle for same tilt.

6.43.1 Detailed Description

Class for handling mobile user input via touch screen and sensors. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.

6.43.2 Member Enumeration Documentation

6.43.2.1 InputType

```
enum NWH.VehiclePhysics.MobileInputManager.InputType [strong]
```

Steer input device. Accelerometer - uses sensors to get horizontal axis. Screen - uses left side of the screen along with touch input to get steering position. `Steering Wheel` - uses `SteeringWheel` script and steering wheel on-screen graphic that can be rotated by dragging.

Enumerator

Accelerometer	
SteeringWheel	

6.43.3 Member Function Documentation

6.43.3.1 BrakeDown()

```
void NWH.VehiclePhysics.MobileInputManager.BrakeDown ( )
```

6.43.3.2 BrakeUp()

```
void NWH.VehiclePhysics.MobileInputManager.BrakeUp ( )
```

6.43.3.3 ChangeCamera()

```
void NWH.VehiclePhysics.MobileInputManager.ChangeCamera ( )
```

6.43.3.4 ChangeVehicle()

```
void NWH.VehiclePhysics.MobileInputManager.ChangeVehicle ( )
```

6.43.3.5 EngineStartStop()

```
void NWH.VehiclePhysics.MobileInputManager.EngineStartStop ( )
```

6.43.3.6 ThrottleDown()

```
void NWH.VehiclePhysics.MobileInputManager.ThrottleDown ( )
```

6.43.3.7 ThrottleUp()

```
void NWH.VehiclePhysics.MobileInputManager.ThrottleUp ( )
```

6.43.3.8 ToggleHighBeams()

```
void NWH.VehiclePhysics.MobileInputManager.ToggleHighBeams ( )
```

6.43.3.9 ToggleLowBeams()

```
void NWH.VehiclePhysics.MobileInputManager.ToggleLowBeams ( )
```

6.43.3.10 TrailerAttachDetach()

```
void NWH.VehiclePhysics.MobileInputManager.TrailerAttachDetach ( )
```

6.43.3.11 UpdateSteering()

```
void NWH.VehiclePhysics.MobileInputManager.UpdateSteering ( )
```

6.43.4 Member Data Documentation

6.43.4.1 inputType

```
InputType NWH.VehiclePhysics.MobileInputManager.inputType = InputType.SteeringWheel
```

Active steer devices.

6.43.4.2 steeringWheel

```
SteeringWheel NWH.VehiclePhysics.MobileInputManager.steeringWheel
```

[Steering](#) wheel script. Optional and not needed if [SteeringWheel](#) option is not used.

6.43.4.3 tiltSensitivity

```
float NWH.VehiclePhysics.MobileInputManager.tiltSensitivity = 1.5f
```

Higher value will result in higher steer angle for same tilt.

6.43.4.4 vehicleChanger

```
VehicleChanger NWH.VehiclePhysics.MobileInputManager.vehicleChanger
```

Set to null (none) if you want to use your own vehicle controller. If this is set to other than null current active vehicle according to the assigned vehicle changer will be used instead of the assigned vehicle controller.

6.43.4.5 vehicleController

`VehicleController` `NWH.VehiclePhysics.MobileInputManager.vehicleController`

If you want to use this script with a single vehicle or want to set your own vehicle controller from script set vehicle changer field to null / none.

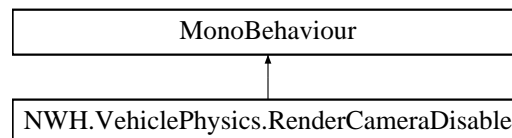
The documentation for this class was generated from the following file:

- `E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/MobileInputManager.cs`

6.44 NWH.VehiclePhysics.RenderCameraDisable Class Reference

Attach this to any cameras rendering to mirrors so that when vehicle is inactive mirror will not update.

Inheritance diagram for `NWH.VehiclePhysics.RenderCameraDisable`:



6.44.1 Detailed Description

Attach this to any cameras rendering to mirrors so that when vehicle is inactive mirror will not update.

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

- `E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/RenderCameraDisable.cs`

6.45 NWH.VehiclePhysics.Rigging Class Reference

Public Member Functions

- void `Update` (`VehicleController` vc)

Public Attributes

- bool `enabled` = false
Should rigging be used?
- bool `solidAxle` = false
If enabled wheel camber will be set as if the wheels were on a solid axle.
- List< Transform > `axleBones` = new List<Transform>()
List of handles controlling the axle bones. Each item is a single axle handle.
- List< Transform > `wheelBones` = new List<Transform>()
List of handles controlling the wheel bones. Each item is a single wheel bone handle.

6.45.1 Member Function Documentation

6.45.1.1 Update()

```
void NWH.VehiclePhysics.Rigging.Update (
    VehicleController vc )
```

6.45.2 Member Data Documentation

6.45.2.1 axleBones

```
List<Transform> NWH.VehiclePhysics.Rigging.axleBones = new List<Transform>()
```

List of handles controlling the axle bones. Each item is a single axle handle.

6.45.2.2 enabled

```
bool NWH.VehiclePhysics.Rigging.enabled = false
```

Should rigging be used?

6.45.2.3 solidAxle

```
bool NWH.VehiclePhysics.Rigging.solidAxle = false
```

If enabled wheel camber will be set as if the wheels were on a solid axle.

6.45.2.4 wheelBones

```
List<Transform> NWH.VehiclePhysics.Rigging.wheelBones = new List<Transform>()
```

List of handles controlling the wheel bones. Each item is a single wheel bone handle.

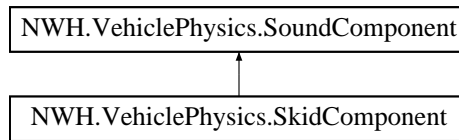
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Rigging.cs](#)

6.46 NWH.VehiclePhysics.SkidComponent Class Reference

[Sound](#) produced by tire skidding over surface.

Inheritance diagram for NWH.VehiclePhysics.SkidComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, AudioManagerGroup amg)
- override void [Update](#) ()

Public Attributes

- float [forwardSkidVolume](#) = 0.8f
Volume of longitudinal (forward) slip / wheel spin sound effect.
- float [sideSkidVolume](#) = 0.9f
Volume of lateral (side) slip sound effect.

Additional Inherited Members

6.46.1 Detailed Description

[Sound](#) produced by tire skidding over surface.

6.46.2 Member Function Documentation

6.46.2.1 Initialize()

```
override void NWH.VehiclePhysics.SkidComponent.Initialize (  
    VehicleController vc,  
    AudioManagerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.46.2.2 Update()

```
override void NWH.VehiclePhysics.SkidComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.46.3 Member Data Documentation

6.46.3.1 forwardSkidVolume

```
float NWH.VehiclePhysics.SkidComponent.forwardSkidVolume = 0.8f
```

Volume of longitudinal (forward) slip / wheel spin sound effect.

6.46.3.2 sideSkidVolume

```
float NWH.VehiclePhysics.SkidComponent.sideSkidVolume = 0.9f
```

Volume of lateral (side) slip sound effect.

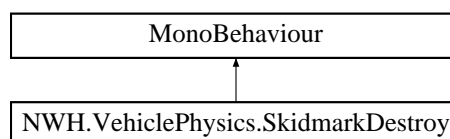
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[SkidComponent.cs](#)

6.47 NWH.VehiclePhysics.SkidmarkDestroy Class Reference

Destroys skidmark object when distance to the vehicle becomes greater then distance threshold.

Inheritance diagram for NWH.VehiclePhysics.SkidmarkDestroy:



Public Attributes

- [VehicleController](#) parentVehicleController
- float [distanceThreshold](#) = 100f

6.47.1 Detailed Description

Destroys skidmark object when distance to the vehicle becomes greater then distance threshold.

6.47.2 Member Data Documentation

6.47.2.1 distanceThreshold

```
float NWH.VehiclePhysics.SkidmarkDestroy.distanceThreshold = 100f
```

6.47.2.2 parentVehicleController

```
VehicleController NWH.VehiclePhysics.SkidmarkDestroy.parentVehicleController
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/[SkidmarkDestroy.cs](#)

6.48 NWH.VehiclePhysics.SkidmarkGenerator Class Reference

Generates skidmark meshes.

Public Member Functions

- void [CreateNewSnapshot](#) (bool generateNew=false)
- void [Initialize](#) ([VehicleController](#) vc, [Wheel](#) wheel)
- void [Update](#) ()
- void [SubArray](#) (ref int[] data, out int[] result, int index, int length)
- void [DoubleSubArray](#) (ref int[] data, out int[] result, int index1, int index2, int length1, int length2)

Public Attributes

- int [maxMarks](#) = 512
- int [maxTris](#)
- float [markWidth](#) = 0.35f
- float [groundOffset](#) = 0.014f
- float [minSqrDistance](#)
- [Wheel](#) wheel

6.48.1 Detailed Description

Generates skidmark meshes.

6.48.2 Member Function Documentation

6.48.2.1 CreateNewSnapshot()

```
void NWH.VehiclePhysics.SkidmarkGenerator.CreateNewSnapshot (
    bool generateNew = false )
```

6.48.2.2 DoubleSubArray()

```
void NWH.VehiclePhysics.SkidmarkGenerator.DoubleSubArray (
    ref int [] data,
    out int [] result,
    int index1,
    int index2,
    int length1,
    int length2 )
```

6.48.2.3 Initialize()

```
void NWH.VehiclePhysics.SkidmarkGenerator.Initialize (
    VehicleController vc,
    Wheel wheel )
```

6.48.2.4 SubArray()

```
void NWH.VehiclePhysics.SkidmarkGenerator.SubArray (
    ref int [] data,
    out int [] result,
    int index,
    int length )
```

6.48.2.5 Update()

```
void NWH.VehiclePhysics.SkidmarkGenerator.Update ( )
```

6.48.3 Member Data Documentation

6.48.3.1 groundOffset

```
float NWH.VehiclePhysics.SkidmarkGenerator.groundOffset = 0.014f
```

6.48.3.2 markWidth

```
float NWH.VehiclePhysics.SkidmarkGenerator.markWidth = 0.35f
```

6.48.3.3 maxMarks

```
int NWH.VehiclePhysics.SkidmarkGenerator.maxMarks = 512
```

6.48.3.4 maxTris

```
int NWH.VehiclePhysics.SkidmarkGenerator.maxTris
```

6.48.3.5 minSqrDistance

```
float NWH.VehiclePhysics.SkidmarkGenerator.minSqrDistance
```

6.48.3.6 wheel

```
Wheel NWH.VehiclePhysics.SkidmarkGenerator.wheel
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/[SkidmarkGenerator.cs](#)

6.49 NWH.VehiclePhysics.Skidmarks Class Reference

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()

Public Attributes

- bool [enabled](#) = true
Should skidmarks be generated?
- float [skidmarkStrength](#) = 0.5f
Higher value will give darker skidmarks for the same skid value.
- float [maxSkidmarkAlpha](#) = 0.6f
Max skidmark texture alpha.
- bool [persistentSkidmarks](#) = false
If enabled skidmarks will stay on the ground until distance from the vehicle becomes greater than persistent↔ SkidmarkDistance. If disabled skidmarks will stay on the ground until maxMarksPerSection is reached and then will start getting deleted from the end.
- float [persistentSkidmarkDistance](#) = 100f
Persistent skidmarks get deleted when distance from the parent vehicle is higher than this.
- int [maxMarksPerSection](#) = 120
Number of skidmarks that will be drawn per one section, before mesh is saved and new one is generated.
- float [minDistance](#) = 0.14f
Distance from the last skidmark section needed to generate a new one.

6.49.1 Member Function Documentation

6.49.1.1 Initialize()

```
void NWH.VehiclePhysics.Skidmarks.Initialize (
    VehicleController vc )
```

6.49.1.2 Update()

```
void NWH.VehiclePhysics.Skidmarks.Update ( )
```

6.49.2 Member Data Documentation

6.49.2.1 enabled

```
bool NWH.VehiclePhysics.Skidmarks.enabled = true
```

Should skidmarks be generated?

6.49.2.2 maxMarksPerSection

```
int NWH.VehiclePhysics.Skidmarks.maxMarksPerSection = 120
```

Number of skidmarks that will be drawn per one section, before mesh is saved and new one is generated.

6.49.2.3 maxSkidmarkAlpha

```
float NWH.VehiclePhysics.Skidmarks.maxSkidmarkAlpha = 0.6f
```

Max skidmark texture alpha.

6.49.2.4 minDistance

```
float NWH.VehiclePhysics.Skidmarks.minDistance = 0.14f
```

Distance from the last skidmark section needed to generate a new one.

6.49.2.5 persistentSkidmarkDistance

```
float NWH.VehiclePhysics.Skidmarks.persistentSkidmarkDistance = 100f
```

Persistent skidmarks get deleted when distance from the parent vehicle is higher than this.

6.49.2.6 persistentSkidmarks

```
bool NWH.VehiclePhysics.Skidmarks.persistentSkidmarks = false
```

If enabled skidmarks will stay on the ground until distance from the vehicle becomes greater than persistentSkidmarkDistance. If disabled skidmarks will stay on the ground until maxMarksPerSection is reached and then will start getting deleted from the end.

6.49.2.7 skidmarkStrength

```
float NWH.VehiclePhysics.Skidmarks.skidmarkStrength = 0.5f
```

Higher value will give darker skidmarks for the same skid value.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/[Skidmarks.cs](#)

6.50 NWH.VehiclePhysics.Sound Class Reference

Main class that manages all the sound aspects of the vehicle.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()
- void [SetAudioSourceDefaults](#) ([AudioSource](#) audioSource, bool play=false, bool loop=false, float volume=0f, [AudioClip](#) clip=null)
Initializes audio source to it's starting values.
- void [Enable](#) ()
Enable sound.
- void [Disable](#) ()
Disable all sound components.
- void [SetDefaults](#) ()
Sets defaults to all the basic sound components when script is first added or reset is called.

Public Attributes

- float [spatialBlend](#) = 1f
Spatial blend of all audio sources. Can not be changed at runtime.
- float [masterVolume](#) = 1f
Master volume of a vehicle. To adjust volume of all vehicles or their components check audio mixer.
- [EngineIdleSoundComponent](#) engineIdleComponent = new [EngineIdleSoundComponent](#)()
Sound of engine idling.
- [EngineStartStopComponent](#) engineStartStopComponent = new [EngineStartStopComponent](#)()
Engine start / stop component. First clip is for starting and second one is for stopping.
- [TurboWhistleComponent](#) turboWhistleComponent = new [TurboWhistleComponent](#)()
Forced induction whistle component. Can be used for air intake noise or supercharger if spool up time is set to 0 under engine settings.
- [TurboFlutterComponent](#) turboFlutterComponent = new [TurboFlutterComponent](#)()
Sound of turbo's wastegate. Supports multiple clips.
- [BackfireComponent](#) exhaustPopComponent = new [BackfireComponent](#)()
Exhaust popping sound on deceleration / rev limiter.
- [TransmissionWhineComponent](#) transmissionWhineComponent = new [TransmissionWhineComponent](#)()
Transmission whine from straight cut gears or just a noisy gearbox.
- [GearChangeComponent](#) gearChangeComponent = new [GearChangeComponent](#)()

- *Sound from changing gears. Supports multiple clips.*
 • `SuspensionComponent suspensionComponent = new SuspensionComponent()`
- *Sound from wheels hitting ground and/or obstacles. Supports multiple clips.*
 • `SurfaceComponent surfaceComponent = new SurfaceComponent()`
- *Sound produced by wheel rolling over a surface. Tire hum.*
 • `SkidComponent skidComponent = new SkidComponent()`
- *Sound produced by wheel skidding over a surface. Tire squeal.*
 • `CrashComponent crashComponent = new CrashComponent()`
- *Sound of vehicle hitting other objects. Supports multiple clips.*
 • `AirBrakeComponent airBrakeComponent = new AirBrakeComponent()`
- *Sound of air brakes releasing air. Supports multiple clips.*
 • `BlinkerComponent blinkerComponent = new BlinkerComponent()`
- *Tick-tock sound of a working blinker. First clip is played when blinker is turning on and second clip is played when blinker is turning off.*
 • `HornComponent hornComponent = new HornComponent()`
- `bool insideVehicle = false`
 • *Set to true if listener inside vehicle. Mixer must be set up.*
- `float interiorAttenuation = -7f`
 • *Sound attenuation inside vehicle.*
- `float lowPassFrequency = 6000f`
- `float lowPassQ = 1f`
- `AudioMixerGroup masterGroup`
- `AudioMixerGroup engineMixerGroup`
- `AudioMixerGroup transmissionMixerGroup`
- `AudioMixerGroup surfaceNoiseMixerGroup`
- `AudioMixerGroup turboMixerGroup`
- `AudioMixerGroup suspensionMixerGroup`
- `AudioMixerGroup crashMixerGroup`
- `AudioMixerGroup otherMixerGroup`

6.50.1 Detailed Description

Main class that manages all the sound aspects of the vehicle.

6.50.2 Member Function Documentation

6.50.2.1 Disable()

```
void NWH.VehiclePhysics.Sound.Disable ( )
```

Disable all sound components.

6.50.2.2 Enable()

```
void NWH.VehiclePhysics.Sound.Enable ( )
```

Enable sound.

6.50.2.3 Initialize()

```
void NWH.VehiclePhysics.Sound.Initialize (
    VehicleController vc )
```

6.50.2.4 SetAudioSourceDefaults()

```
void NWH.VehiclePhysics.Sound.SetAudioSourceDefaults (
    AudioSource audioSource,
    bool play = false,
    bool loop = false,
    float volume = 0f,
    AudioClip clip = null )
```

Initializes audio source to it's starting values.

Parameters

<i>audioSource</i>	AudioSource in question.
<i>play</i>	Play on awake?
<i>loop</i>	Should clip be looped?
<i>volume</i>	Volume of the audio source.
<i>clip</i>	Clip that will be set at the start.

6.50.2.5 SetDefaults()

```
void NWH.VehiclePhysics.Sound.SetDefaults ( )
```

Sets defaults to all the basic sound components when script is first added or reset is called.

6.50.2.6 Update()

```
void NWH.VehiclePhysics.Sound.Update ( )
```

6.50.3 Member Data Documentation

6.50.3.1 airBrakeComponent

`AirBrakeComponent` NWH.VehiclePhysics.Sound.airBrakeComponent = new `AirBrakeComponent`()

Sound of air brakes releasing air. Supports multiple clips.

6.50.3.2 blinkerComponent

`BlinkerComponent` NWH.VehiclePhysics.Sound.blinkerComponent = new `BlinkerComponent`()

Tick-tock sound of a working blinker. First clip is played when blinker is turning on and second clip is played when blinker is turning off.

6.50.3.3 crashComponent

`CrashComponent` NWH.VehiclePhysics.Sound.crashComponent = new `CrashComponent`()

Sound of vehicle hitting other objects. Supports multiple clips.

6.50.3.4 crashMixerGroup

`AudioMixerGroup` NWH.VehiclePhysics.Sound.crashMixerGroup

6.50.3.5 engineIdleComponent

`EngineIdleSoundComponent` NWH.VehiclePhysics.Sound.engineIdleComponent = new `EngineIdleSoundComponent`()

Sound of engine idling.

6.50.3.6 engineMixerGroup

`AudioMixerGroup` NWH.VehiclePhysics.Sound.engineMixerGroup

6.50.3.7 engineStartStopComponent

```
EngineStartStopComponent NWH.VehiclePhysics.Sound.engineStartStopComponent = new EngineStartStopComponent()
```

[Engine](#) start / stop component. First clip is for starting and second one is for stopping.

6.50.3.8 exhaustPopComponent

```
BackfireComponent NWH.VehiclePhysics.Sound.exhaustPopComponent = new BackfireComponent()
```

Exhaust popping sound on deceleration / rev limiter.

6.50.3.9 gearChangeComponent

```
GearChangeComponent NWH.VehiclePhysics.Sound.gearChangeComponent = new GearChangeComponent()
```

[Sound](#) from changing gears. Supports multiple clips.

6.50.3.10 hornComponent

```
HornComponent NWH.VehiclePhysics.Sound.hornComponent = new HornComponent()
```

6.50.3.11 insideVehicle

```
bool NWH.VehiclePhysics.Sound.insideVehicle = false
```

Set to true if listener inside vehicle. Mixer must be set up.

6.50.3.12 interiorAttenuation

```
float NWH.VehiclePhysics.Sound.interiorAttenuation = -7f
```

[Sound](#) attenuation inside vehicle.

6.50.3.13 lowPassFrequency

```
float NWH.VehiclePhysics.Sound.lowPassFrequency = 6000f
```

6.50.3.14 lowPassQ

```
float NWH.VehiclePhysics.Sound.lowPassQ = 1f
```

6.50.3.15 masterGroup

```
AudioMixerGroup NWH.VehiclePhysics.Sound.masterGroup
```

6.50.3.16 masterVolume

```
float NWH.VehiclePhysics.Sound.masterVolume = 1f
```

Master volume of a vehicle. To adjust volume of all vehicles or their components check audio mixer.

6.50.3.17 otherMixerGroup

```
AudioMixerGroup NWH.VehiclePhysics.Sound.otherMixerGroup
```

6.50.3.18 skidComponent

```
SkidComponent NWH.VehiclePhysics.Sound.skidComponent = new SkidComponent()
```

[Sound](#) produced by wheel skidding over a surface. Tire squeal.

6.50.3.19 spatialBlend

```
float NWH.VehiclePhysics.Sound.spatialBlend = 1f
```

Spatial blend of all audio sources. Can not be changed at runtime.

6.50.3.20 surfaceComponent

```
SurfaceComponent NWH.VehiclePhysics.Sound.surfaceComponent = new SurfaceComponent ()
```

Sound produced by wheel rolling over a surface. Tire hum.

6.50.3.21 surfaceNoiseMixerGroup

```
AudioMixerGroup NWH.VehiclePhysics.Sound.surfaceNoiseMixerGroup
```

6.50.3.22 suspensionComponent

```
SuspensionComponent NWH.VehiclePhysics.Sound.suspensionComponent = new SuspensionComponent ()
```

Sound from wheels hitting ground and/or obstracles. Supports multiple clips.

6.50.3.23 suspensionMixerGroup

```
AudioMixerGroup NWH.VehiclePhysics.Sound.suspensionMixerGroup
```

6.50.3.24 transmissionMixerGroup

```
AudioMixerGroup NWH.VehiclePhysics.Sound.transmissionMixerGroup
```

6.50.3.25 transmissionWhineComponent

```
TransmissionWhineComponent NWH.VehiclePhysics.Sound.transmissionWhineComponent = new TransmissionWhineComponent ()
```

Transmission whine from straight cut gears or just a noisy gearbox.

6.50.3.26 turboFlutterComponent

```
TurboFlutterComponent NWH.VehiclePhysics.Sound.turboFlutterComponent = new TurboFlutterComponent ()
```

Sound of turbo's wastegate. Supports multiple clips.

6.50.3.27 turboMixerGroup

AudioMixerGroup NWH.VehiclePhysics.Sound.turboMixerGroup

6.50.3.28 turboWhistleComponent

`TurboWhistleComponent` NWH.VehiclePhysics.Sound.turboWhistleComponent = new `TurboWhistleComponent`()

Forced induction whistle component. Can be used for air intake noise or supercharger if spool up time is set to 0 under engine settings.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[Sound.cs](#)

6.51 NWH.VehiclePhysics.GroundDetection.SoundComponent Class Reference

Base class for surface sounds.

Public Attributes

- float `volume` = 0.6f
- float `pitch` = 1f
- AudioClip `clip`
- AudioSource `source`

6.51.1 Detailed Description

Base class for surface sounds.

6.51.2 Member Data Documentation

6.51.2.1 clip

AudioClip NWH.VehiclePhysics.GroundDetection.SoundComponent.clip

6.51.2.2 pitch

```
float NWH.VehiclePhysics.GroundDetection.SoundComponent.pitch = 1f
```

6.51.2.3 source

```
AudioSource NWH.VehiclePhysics.GroundDetection.SoundComponent.source
```

6.51.2.4 volume

```
float NWH.VehiclePhysics.GroundDetection.SoundComponent.volume = 0.6f
```

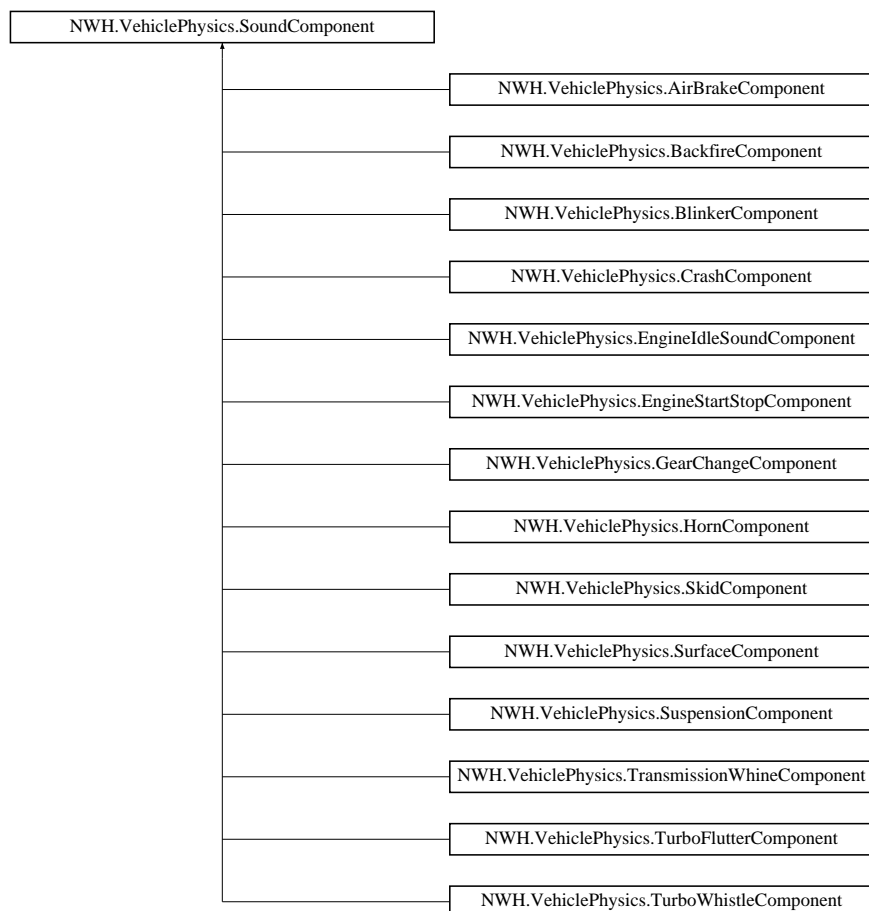
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[GroundDetection.cs](#)

6.52 NWH.VehiclePhysics.SoundComponent Class Reference

Base abstract class from which all vehicle sound components inherit.

Inheritance diagram for NWH.VehiclePhysics.SoundComponent:



Public Member Functions

- void [RegisterSources](#) ()
Adds outputs of sources to the mixer.
- void [SetVolume](#) (float [volume](#), int id)
Sets volume for the [id]th source in sources list. Use instead of directly changing source volume as this takes master volume into account.
- void [SetVolume](#) (float [volume](#))
Sets volume for the first source in sources list. Use instead of directly changing source volume as this takes master volume into account.
- void [Enable](#) ()
- void [Disable](#) ()
- abstract void [Initialize](#) ([VehicleController](#) vc, AudioManagerGroup amg)
- abstract void [Update](#) ()

Public Attributes

- float [volume](#) = 0.1f
Base volume of the sound component.
- float [pitch](#) = 1f
Base pitch of the sound component.
- List< AudioClip > [clips](#) = new List<AudioClip>()
List of audio clips this component can use. Some components can use multiple clips in which case they will be chosen at random, and some components can use only one in which case only the first clip will be selected. Check manual for more details.

Protected Attributes

- List< AudioSource > [sources](#) = new List<AudioSource>()
- [VehicleController](#) vc
- AudioManagerGroup [audioMixerGroup](#)

Properties

- AudioClip [Clip](#) [get, set]
Gets or sets the first clip in the clips list.
- List< AudioClip > [Clips](#) [get, set]
Gets or sets the whole clip list.
- AudioSource [Source](#) [get, set]
Gets or sets the first audio source in the sources list.
- List< AudioSource > [Sources](#) [get, set]
Gets or sets the whole sources list.
- AudioClip [RandomClip](#) [get]
Gets a random clip from clips list.

6.52.1 Detailed Description

Base abstract class from which all vehicle sound components inherit.

6.52.2 Member Function Documentation

6.52.2.1 Disable()

```
void NWH.VehiclePhysics.SoundComponent.Disable ( )
```

6.52.2.2 Enable()

```
void NWH.VehiclePhysics.SoundComponent.Enable ( )
```

6.52.2.3 Initialize()

```
abstract void NWH.VehiclePhysics.SoundComponent.Initialize (
    VehicleController vc,
    AudioMixerGroup amg ) [pure virtual]
```

Implemented in [NWH.VehiclePhysics.EngineIdleSoundComponent](#), [NWH.VehiclePhysics.SkidComponent](#), [NWH.VehiclePhysics.TransmissionWhineComponent](#), [NWH.VehiclePhysics.TurboWhistleComponent](#), [NWH.VehiclePhysics.BlinkerComponent](#), [NWH.VehiclePhysics.GearChangeComponent](#), [NWH.VehiclePhysics.AirBrakeComponent](#), [NWH.VehiclePhysics.SuspensionComponent](#), [NWH.VehiclePhysics.CrashComponent](#), [NWH.VehiclePhysics.BackfireComponent](#), [NWH.VehiclePhysics.EngineStartStopComponent](#), [NWH.VehiclePhysics.HornComponent](#), [NWH.VehiclePhysics.SurfaceComponent](#), and [NWH.VehiclePhysics.TurboFlutterComponent](#).

6.52.2.4 RegisterSources()

```
void NWH.VehiclePhysics.SoundComponent.RegisterSources ( )
```

Adds outputs of sources to the mixer.

6.52.2.5 SetVolume() [1/2]

```
void NWH.VehiclePhysics.SoundComponent.SetVolume (
    float volume,
    int id )
```

Sets volume for the [id]th source in sources list. Use instead of directly changing source volume as this takes master volume into account.

6.52.2.6 SetVolume() [2/2]

```
void NWH.VehiclePhysics.SoundComponent.SetVolume (
    float volume )
```

Sets volume for the first source in sources list. Use instead of directly changing source volume as this takes master volume into account.

6.52.2.7 Update()

```
abstract void NWH.VehiclePhysics.SoundComponent.Update ( ) [pure virtual]
```

Implemented in [NWH.VehiclePhysics.EngineIdleSoundComponent](#), [NWH.VehiclePhysics.SkidComponent](#), [NWH.VehiclePhysics.SuspensionComponent](#), [NWH.VehiclePhysics.TransmissionWhineComponent](#), [NWH.VehiclePhysics.TurboWhistleComponent](#), [NWH.VehiclePhysics.GearChangeComponent](#), [NWH.VehiclePhysics.BlinkerComponent](#), [NWH.VehiclePhysics.SurfaceComponent](#), [NWH.VehiclePhysics.AirBrakeComponent](#), [NWH.VehiclePhysics.CrashComponent](#), [NWH.VehiclePhysics.EngineStartStopComponent](#), [NWH.VehiclePhysics.BackfireComponent](#), [NWH.VehiclePhysics.HornComponent](#), and [NWH.VehiclePhysics.TurboFlutterComponent](#).

6.52.3 Member Data Documentation

6.52.3.1 audioMixerGroup

```
AudioMixerGroup NWH.VehiclePhysics.SoundComponent.audioMixerGroup [protected]
```

6.52.3.2 clips

```
List<AudioClip> NWH.VehiclePhysics.SoundComponent.clips = new List<AudioClip>()
```

List of audio clips this component can use. Some components can use multiple clips in which case they will be chosen at random, and some components can use only one in which case only the first clip will be selected. Check manual for more details.

6.52.3.3 pitch

```
float NWH.VehiclePhysics.SoundComponent.pitch = 1f
```

Base pitch of the sound component.

6.52.3.4 sources

```
List<AudioSource> NWH.VehiclePhysics.SoundComponent.sources = new List<AudioSource>() [protected]
```

6.52.3.5 vc

```
VehicleController NWH.VehiclePhysics.SoundComponent.vc [protected]
```

6.52.3.6 volume

```
float NWH.VehiclePhysics.SoundComponent.volume = 0.1f
```

Base volume of the sound component.

6.52.4 Property Documentation

6.52.4.1 Clip

```
AudioClip NWH.VehiclePhysics.SoundComponent.Clip [get], [set]
```

Gets or sets the first clip in the clips list.

6.52.4.2 Clips

```
List<AudioClip> NWH.VehiclePhysics.SoundComponent.Clips [get], [set]
```

Gets or sets the whole clip list.

6.52.4.3 RandomClip

```
AudioClip NWH.VehiclePhysics.SoundComponent.RandomClip [get]
```

Gets a random clip from clips list.

6.52.4.4 Source

AudioSource NWH.VehiclePhysics.SoundComponent.Source [get], [set]

Gets or sets the first audio source in the sources list.

6.52.4.5 Sources

List<AudioSource> NWH.VehiclePhysics.SoundComponent.Sources [get], [set]

Gets or sets the whole sources list.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[SoundComponent.cs](#)

6.53 NWH.WheelController3D.WheelController.Spring Class Reference

Suspension part.

Public Attributes

- float [maxLength](#) = 0.3f
- AnimationCurve [forceCurve](#) = null
- float [maxForce](#) = 22000.0f
- float [length](#)
- float [prevLength](#)
- float [compressionPercent](#)
- float [force](#)
- float [velocity](#)
- Vector3 [targetPoint](#)
- float [overflow](#)
- float [prevOverflow](#)
- float [overflowVelocity](#)
- float [bottomOutForce](#)
- bool [bottomedOut](#)
- bool [overExtended](#)

6.53.1 Detailed Description

Suspension part.

6.53.2 Member Data Documentation

6.53.2.1 bottomedOut

```
bool NWH.WheelController3D.WheelController.Spring.bottomedOut
```

6.53.2.2 bottomOutForce

```
float NWH.WheelController3D.WheelController.Spring.bottomOutForce
```

6.53.2.3 compressionPercent

```
float NWH.WheelController3D.WheelController.Spring.compressionPercent
```

6.53.2.4 force

```
float NWH.WheelController3D.WheelController.Spring.force
```

6.53.2.5 forceCurve

```
AnimationCurve NWH.WheelController3D.WheelController.Spring.forceCurve = null
```

6.53.2.6 length

```
float NWH.WheelController3D.WheelController.Spring.length
```

6.53.2.7 maxForce

```
float NWH.WheelController3D.WheelController.Spring.maxForce = 22000.0f
```

6.53.2.8 maxLength

```
float NWH.WheelController3D.WheelController.Spring.maxLength = 0.3f
```

6.53.2.9 overExtended

`bool NWH.WheelController3D.WheelController.Spring.overExtended`

6.53.2.10 overflow

`float NWH.WheelController3D.WheelController.Spring.overflow`

6.53.2.11 overflowVelocity

`float NWH.WheelController3D.WheelController.Spring.overflowVelocity`

6.53.2.12 prevLength

`float NWH.WheelController3D.WheelController.Spring.prevLength`

6.53.2.13 prevOverflow

`float NWH.WheelController3D.WheelController.Spring.prevOverflow`

6.53.2.14 targetPoint

`Vector3 NWH.WheelController3D.WheelController.Spring.targetPoint`

6.53.2.15 velocity

`float NWH.WheelController3D.WheelController.Spring.velocity`

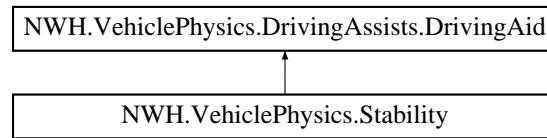
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/[WheelController.cs](#)

6.54 NWH.VehiclePhysics.Stability Class Reference

Traction control class.

Inheritance diagram for NWH.VehiclePhysics.Stability:



Public Member Functions

- void [Update](#) ([VehicleController](#) vc)

Additional Inherited Members

6.54.1 Detailed Description

Traction control class.

6.54.2 Member Function Documentation

6.54.2.1 Update()

```
void NWH.VehiclePhysics.Stability.Update (  
    VehicleController vc )
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/[Stability.cs](#)

6.55 NWH.VehiclePhysics.Steering Class Reference

Everything related to steering and axle's geometry.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Steer](#) ()
- void [AdjustGeometry](#) ()

Public Attributes

- float `lowSpeedAngle` = 30f
Steering angle at low speeds.
- float `highSpeedAngle` = 14f
Steering angle at high speeds.
- float `crossoverSpeed` = 35f
Speed after which only high speed angle will be used. Also affects dynamic smoothing.
- float `degreesPerSecondLimit` = 250f
Only used if `limitSteeringRate` is true. Will limit wheels so that they can only steer up to the set degree limit per second. E.g. 60 degrees per second will mean that the wheels that have 30 degree steer angle will take 1 second to steer from full left to full right.
- float `steeringWheelTurnRatio` = 2f
Steer angle will be multiplied by this value to get steering wheel angle. Ignored if steering wheel is null.
- GameObject `steeringWheel`
[Optional] Steering wheel game object.
- float `trackedSteerIntensity` = 1f
Intensity of braking used when steering a tracked vehicle. Percentage of the max brake torque. Set to 1 for 100% braking torque when using steer. Higer value will make the tracked vehicle turn tighter but will slow it down more.

Properties

- float `Angle` [get]
Current steer angle.

6.55.1 Detailed Description

Everything related to steering and axle's geometry.

6.55.2 Member Function Documentation

6.55.2.1 AdjustGeometry()

```
void NWH.VehiclePhysics.Steering.AdjustGeometry ( )
```

6.55.2.2 Initialize()

```
void NWH.VehiclePhysics.Steering.Initialize (
    VehicleController vc )
```

6.55.2.3 Steer()

```
void NWH.VehiclePhysics.Steering.Steer ( )
```

6.55.3 Member Data Documentation

6.55.3.1 crossoverSpeed

```
float NWH.VehiclePhysics.Steering.crossoverSpeed = 35f
```

Speed after which only high speed angle will be used. Also affects dynamic smoothing.

6.55.3.2 degreesPerSecondLimit

```
float NWH.VehiclePhysics.Steering.degreesPerSecondLimit = 250f
```

Only used if limitSteeringRate is true. Will limit wheels so that they can only steer up to the set degree limit per second. E.g. 60 degrees per second will mean that the wheels that have 30 degree steer angle will take 1 second to steer from full left to full right.

6.55.3.3 highSpeedAngle

```
float NWH.VehiclePhysics.Steering.highSpeedAngle = 14f
```

[Steering](#) angle at high speeds.

6.55.3.4 lowSpeedAngle

```
float NWH.VehiclePhysics.Steering.lowSpeedAngle = 30f
```

[Steering](#) angle at low speeds.

6.55.3.5 steeringWheel

```
GameObject NWH.VehiclePhysics.Steering.steeringWheel
```

[Optional] [Steering](#) wheel game object.

6.55.3.6 steeringWheelTurnRatio

```
float NWH.VehiclePhysics.Steering.steeringWheelTurnRatio = 2f
```

Steer angle will be multiplied by this value to get steering wheel angle. Ignored if steering wheel is null.

6.55.3.7 trackedSteerIntensity

```
float NWH.VehiclePhysics.Steering.trackedSteerIntensity = 1f
```

Intensity of braking used when steering a tracked vehicle. Percentage of the max brake torque. Set to 1 for 100% braking torque when using steer. Higer value will make the tracked vehicle turn tighter but will slow it down more.

6.55.4 Property Documentation

6.55.4.1 Angle

```
float NWH.VehiclePhysics.Steering.Angle [get]
```

Current steer angle.

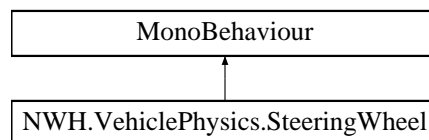
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Steering.cs](#)

6.56 NWH.VehiclePhysics.SteeringWheel Class Reference

Script for controlling the GUI steering wheel for mobile controls. Credits go to yasirkula from Unity Forums for original code.

Inheritance diagram for NWH.VehiclePhysics.SteeringWheel:



Public Member Functions

- float [GetClampedValue](#) ()
Returns a value in range [-1,1] similar to GetAxis("Horizontal")
- void [PressEvent](#) (BaseEventData eventData)
- void [DragEvent](#) (BaseEventData eventData)
- void [ReleaseEvent](#) (BaseEventData eventData)

Public Attributes

- Graphic [steeringWheelGraphic](#)
- float [maximumSteeringAngle](#) = 200f
Maximum angle that the steering wheel can be turned to towards either side in degrees.
- float [returnToCenterSpeed](#) = 400f
Speed at which wheel is returned to center in degrees per second.

6.56.1 Detailed Description

Script for controlling the GUI steering wheel for mobile controls. Credits go to yasirkula from Unity Forums for original code.

6.56.2 Member Function Documentation

6.56.2.1 DragEvent()

```
void NWH.VehiclePhysics.SteeringWheel.DragEvent (
    BaseEventData eventData )
```

6.56.2.2 GetClampedValue()

```
float NWH.VehiclePhysics.SteeringWheel.GetClampedValue ( )
```

Returns a value in range [-1,1] similar to GetAxis("Horizontal")

6.56.2.3 PressEvent()

```
void NWH.VehiclePhysics.SteeringWheel.PressEvent (
    BaseEventData eventData )
```

6.56.2.4 ReleaseEvent()

```
void NWH.VehiclePhysics.SteeringWheel.ReleaseEvent (
    BaseEventData eventData )
```

6.56.3 Member Data Documentation

6.56.3.1 maximumSteeringAngle

```
float NWH.VehiclePhysics.SteeringWheel.maximumSteeringAngle = 200f
```

Maximum angle that the steering wheel can be turned to towards either side in degrees.

6.56.3.2 returnToCenterSpeed

```
float NWH.VehiclePhysics.SteeringWheel.returnToCenterSpeed = 400f
```

Speed at which wheel is returned to center in degrees per second.

6.56.3.3 steeringWheelGraphic

```
Graphic NWH.VehiclePhysics.SteeringWheel.steeringWheelGraphic
```

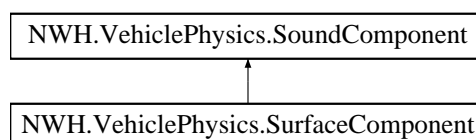
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/[SteeringWheel.cs](#)

6.57 NWH.VehiclePhysics.SurfaceComponent Class Reference

Sounds produced by tire rolling over the surface.

Inheritance diagram for NWH.VehiclePhysics.SurfaceComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, [AudioMixerGroup](#) amg)
- override void [Update](#) ()

Additional Inherited Members

6.57.1 Detailed Description

Sounds produced by tire rolling over the surface.

6.57.2 Member Function Documentation

6.57.2.1 Initialize()

```
override void NWH.VehiclePhysics.SurfaceComponent.Initialize (
    VehicleController vc,
    AudioMixerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.57.2.2 Update()

```
override void NWH.VehiclePhysics.SurfaceComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[SurfaceComponent.cs](#)

6.58 NWH.VehiclePhysics.SurfaceParticles Class Reference

Skid smoke and surface dust generated by wheel slipping / rolling over the surface.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc, [Wheel](#) wheel)
- void [Update](#) ()

Public Attributes

- ParticleSystem [smoke](#)
- ParticleSystem [dust](#)
- GameObject [smokeGo](#)
- GameObject [dustGo](#)

6.58.1 Detailed Description

Skid smoke and surface dust generated by wheel slipping / rolling over the surface.

6.58.2 Member Function Documentation

6.58.2.1 Initialize()

```
void NWH.VehiclePhysics.SurfaceParticles.Initialize (
    VehicleController vc,
    Wheel wheel )
```

6.58.2.2 Update()

```
void NWH.VehiclePhysics.SurfaceParticles.Update ( )
```

6.58.3 Member Data Documentation

6.58.3.1 dust

```
ParticleSystem NWH.VehiclePhysics.SurfaceParticles.dust
```

6.58.3.2 dustGo

```
GameObject NWH.VehiclePhysics.SurfaceParticles.dustGo
```

6.58.3.3 smoke

```
ParticleSystem NWH.VehiclePhysics.SurfaceParticles.smoke
```

6.58.3.4 smokeGo

GameObject NWH.VehiclePhysics.SurfaceParticles.smokeGo

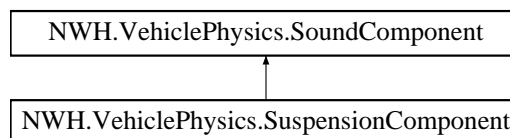
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/[SurfaceParticles.cs](#)

6.59 NWH.VehiclePhysics.SuspensionComponent Class Reference

[Sound](#) of wheel hitting the surface or obstracle.

Inheritance diagram for NWH.VehiclePhysics.SuspensionComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController vc](#), AudioManagerGroup amg)
- override void [Update](#) ()

Additional Inherited Members

6.59.1 Detailed Description

[Sound](#) of wheel hitting the surface or obstracle.

6.59.2 Member Function Documentation

6.59.2.1 Initialize()

```
override void NWH.VehiclePhysics.SuspensionComponent.Initialize (  
    VehicleController vc,  
    AudioManagerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.59.2.2 Update()

```
override void NWH.VehiclePhysics.SuspensionComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[SuspensionComponent.cs](#)

6.60 NWH.VehiclePhysics.Tracks Class Reference

Class for handling tracked vehicles. If enabled all the wheels should be the same size and placed on either left or right side of the vehicle. Center wheels are not supported in this mode.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()

Public Attributes

- bool [trackedVehicle](#) = false
True if vehicle uses tracks instead of wheels.
- float [textureOffsetSpeedCoefficient](#) = 1f
Speed at which the track texture will be moved. Needs to be adjusted to match wheel rotation speed.
- [Renderer](#) [leftTrackRenderer](#)
Renderers for left track. Will be used to offset the track texture to imitate moving tracks.
- [Renderer](#) [rightTrackRenderer](#)
Renderers for right track. Will be used to offset the track texture to imitate moving tracks.
- List< [GameObject](#) > [leftSprockets](#) = new List<[GameObject](#)>()
Left side drive sprocket and other rotating wheels that are not wheel controllers but need to be rotated with track.
- List< [GameObject](#) > [rightSprockets](#) = new List<[GameObject](#)>()
Right side drive sprocket and other rotating wheels that are not wheel controllers but need to be rotated with track.

6.60.1 Detailed Description

Class for handling tracked vehicles. If enabled all the wheels should be the same size and placed on either left or right side of the vehicle. Center wheels are not supported in this mode.

6.60.2 Member Function Documentation

6.60.2.1 Initialize()

```
void NWH.VehiclePhysics.Tracks.Initialize (
    VehicleController vc )
```

6.60.2.2 Update()

```
void NWH.VehiclePhysics.Tracks.Update ( )
```

6.60.3 Member Data Documentation

6.60.3.1 leftSprockets

```
List<GameObject> NWH.VehiclePhysics.Tracks.leftSprockets = new List<GameObject>()
```

Left side drive sprocket and other rotating wheels that are not wheel controllers but need to be rotated with track.

6.60.3.2 leftTrackRenderer

```
Renderer NWH.VehiclePhysics.Tracks.leftTrackRenderer
```

Renderers for left track. Will be used to offset the track texture to imitate moving tracks.

6.60.3.3 rightSprockets

```
List<GameObject> NWH.VehiclePhysics.Tracks.rightSprockets = new List<GameObject>()
```

Right side drive sprocket and other rotating wheels that are not wheel controllers but need to be rotated with track.

6.60.3.4 rightTrackRenderer

```
Renderer NWH.VehiclePhysics.Tracks.rightTrackRenderer
```

Renderers for right track. Will be used to offset the track texture to imitate moving tracks.

6.60.3.5 textureOffsetSpeedCoefficient

```
float NWH.VehiclePhysics.Tracks.textureOffsetSpeedCoefficient = 1f
```

Speed at which the track texture will be moved. Needs to be adjusted to match wheel rotation speed.

6.60.3.6 trackedVehicle

```
bool NWH.VehiclePhysics.Tracks.trackedVehicle = false
```

True if vehicle uses tracks instead of wheels.

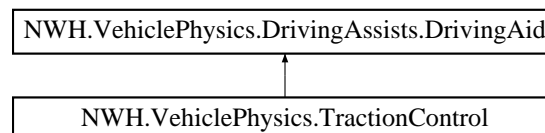
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Tracks.cs](#)

6.61 NWH.VehiclePhysics.TractionControl Class Reference

Traction control class.

Inheritance diagram for NWH.VehiclePhysics.TractionControl:



Public Member Functions

- void [Update](#) ([VehicleController](#) vc)

Public Attributes

- float [powerReduction](#)
Engine power reduction when wheels start slipping.
- float [prevPowerReduction](#)

6.61.1 Detailed Description

Traction control class.

6.61.2 Member Function Documentation

6.61.2.1 Update()

```
void NWH.VehiclePhysics.TractionControl.Update (
    VehicleController vc )
```

6.61.3 Member Data Documentation

6.61.3.1 powerReduction

```
float NWH.VehiclePhysics.TractionControl.powerReduction
```

[Engine](#) power reduction when wheels start slipping.

6.61.3.2 prevPowerReduction

```
float NWH.VehiclePhysics.TractionControl.prevPowerReduction
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/[TractionControl.cs](#)

6.62 NWH.VehiclePhysics.TrailerHandler Class Reference

Everthing related to a trailer.

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()

Public Attributes

- bool `isTrailer` = false
Set to true if the vehicle is a trailer, otherwise set to false.
- GameObject `attachmentPoint` = null
If the vehicle is a trailer, this is the object placed at the point at which it will connect to the towing vehicle. If the vehicle is towing, this is the object placed at point at which trailer will be connected.
- string `trailersTag` = "Trailer"
Tag which will be taken into consideration when searching for a trailer in the scene.
- float `attachDistanceThreshold` = 0.5f
Maximum distance between towing vehicle's attachment point and trailer's attachment point.
- bool `attachOnPlay` = false
If a trailer is in range when the scene is started it will be attached.
- float `breakForce` = Mathf.Infinity
Breaking force of the generated joint.
- bool `trailerInRange` = false
Is trailer's attachment point close enough to be attached to the towing vehicle?
- GameObject `trailerStand`
Object that will be disabled when trailer is attached and disabled when trailer is detached.
- bool `attached`
True if object is trailer and is attached to a towing vehicle and also true if towing vehicle and has trailer attached.
- float `maxNoTrailerPowerReduction` = 0f
Power reduction that will be applied when vehicle has no trailer to avoid wheel spin when controlled with a binary controller.

Properties

- float `NoTrailerPowerReduction` [get]
Return 0 if no trailer and noTrailerPowerReduction value if trailer attached.

6.62.1 Detailed Description

Everything related to a trailer.

6.62.2 Member Function Documentation

6.62.2.1 Initialize()

```
void NWH.VehiclePhysics.TrailerHandler.Initialize (
    VehicleController vc )
```

6.62.2.2 Update()

```
void NWH.VehiclePhysics.TrailerHandler.Update ( )
```

6.62.3 Member Data Documentation

6.62.3.1 attachDistanceThreshold

```
float NWH.VehiclePhysics.TrailerHandler.attachDistanceThreshold = 0.5f
```

Maximum distance between towing vehicle's attachment point and trailer's attachment point.

6.62.3.2 attached

```
bool NWH.VehiclePhysics.TrailerHandler.attached
```

True if object is trailer and is attached to a towing vehicle and also true if towing vehicle and has trailer attached.

6.62.3.3 attachmentPoint

```
GameObject NWH.VehiclePhysics.TrailerHandler.attachmentPoint = null
```

If the vehicle is a trailer, this is the object placed at the point at which it will connect to the towing vehicle. If the vehicle is towing, this is the object placed at point at which trailer will be connected.

6.62.3.4 attachOnPlay

```
bool NWH.VehiclePhysics.TrailerHandler.attachOnPlay = false
```

If a trailer is in range when the scene is started it will be attached.

6.62.3.5 breakForce

```
float NWH.VehiclePhysics.TrailerHandler.breakForce = Mathf.Infinity
```

Breaking force of the generated joint.

6.62.3.6 isTrailer

```
bool NWH.VehiclePhysics.TrailerHandler.isTrailer = false
```

Set to true if the vehicle is a trailer, otherwise set to false.

6.62.3.7 maxNoTrailerPowerReduction

```
float NWH.VehiclePhysics.TrailerHandler.maxNoTrailerPowerReduction = 0f
```

Power reduction that will be applied when vehicle has no trailer to avoid wheel spin when controlled with a binary controller.

6.62.3.8 trailerInRange

```
bool NWH.VehiclePhysics.TrailerHandler.trailerInRange = false
```

Is trailer's attachment point close enough to be attached to the towing vehicle?

6.62.3.9 trailersTag

```
string NWH.VehiclePhysics.TrailerHandler.trailersTag = "Trailer"
```

Tag which will be taken into consideration when searching for a trailer in the scene.

6.62.3.10 trailerStand

```
GameObject NWH.VehiclePhysics.TrailerHandler.trailerStand
```

Object that will be disabled when trailer is attached and disabled when trailer is detached.

6.62.4 Property Documentation

6.62.4.1 NoTrailerPowerReduction

```
float NWH.VehiclePhysics.TrailerHandler.NoTrailerPowerReduction [get]
```

Return 0 if no trailer and noTrailerPowerReduction value if trailer attached.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[TrailerHandler.cs](#)

6.63 NWH.VehiclePhysics.Transmission Class Reference

Handles gear changing and also torque and RPM transmission in both directions.

Public Types

- enum [TransmissionType](#) { [TransmissionType.Manual](#), [TransmissionTypeAutomatic](#), [TransmissionTypeAutomaticSequential](#) }
- enum [DifferentialType](#) { [DifferentialType.Equal](#), [DifferentialType.Open](#), [DifferentialType.LimitedSlip](#), [DifferentialType.Locking](#) }

Public Member Functions

- void [Initialize](#) ([VehicleController](#) vc)
- void [Update](#) ()
- void [UpdateClutch](#) ()
Updates clutch state.
- float [GetMaxSpeedForGear](#) (int g)
Returns maximum speed for the engine's maxRPM and gear ratio.
- float [GetGearRatio](#) (int g)
Total gear ratio of the transmission for the specific gear.
- float [GetClutchEngagementAtPedalPosition](#) (float clutchPercent)
Returns clutch value in range of 1 (disconnected) to 0 (connected) for the passed pedal travel.
- void [ReconstructGearList](#) ()
Recreates gear list from the forward and reverse gears lists.
- void [TorqueSplit](#) (float torque, float topRPM)
Distribute torque between axles.
- void [ShiftInto](#) (int g)
Shifts into given gear. 0 for neutral, less than 0 for reverse and above 0 for forward gears.
- float [TransmitTorque](#) (float inputTorque)
Converts engine torque to axle torque.
- float [ReverseTransmitTorque](#) (float inputTorque)
Converts axle torque to engine torque.
- float [TransmitRPM](#) (float inputRPM)
Converts engine RPM to axle RPM.
- float [ReverseTransmitRPM](#) (float inputRPM)
Converts axle RPM to engine RPM.
- float [ReverseTransmitRPM](#) (float inputRPM, int g)
Converts axle RPM to engine RPM for given gear in Gears list.

Public Attributes

- [TransmissionType transmissionType](#) = [TransmissionTypeAutomaticSequential](#)
Determines in which way gears can be changed. Manual - gears can only be shifted by manual user input. Automatic - automatic gear changing. Allows for gear skipping (e.g. 3rd->5th) which can be useful in trucks and other high gear count vehicles. AutomaticSequential - automatic gear changing but only one gear at the time can be shifted (e.g. 3rd->4th)
- [DifferentialType differentialType](#) = [DifferentialTypeEqual](#)
Differential(s) that determine how the torque will be split between axles. Each axle then has its own differential (check axle settings).
- float [gearMultiplier](#) = 1
Final gear multiplier. Each gear gets multiplied by this value.
- float [targetShiftUpRPM](#) = 3600
RPM at which automatic transmission will shift up. If dynamic shift point is enabled this value will change depending on load.
- float [targetShiftDownRPM](#) = 1400
RPM at which automatic transmission will shift down. If dynamic shift point is enabled this value will change depending on load.
- bool [dynamicShiftPoint](#) = true
If enabled transmission will adjust both shift up and down points to match current load.
- float [shiftPointRandomness](#) = 0.05f
Shift point will randomly vary by the following percent of it's value.
- float [shiftDuration](#) = 0.2f
Time it takes transmission to shift between gears.
- float [shiftDurationRandomness](#) = 0.2f
Maximum percentage of shift duration that will be added or subtracted to it. Default is 20% (0.2f).
- float [postShiftBan](#) = 0.5f
Time after shifting in which shifting can not be done again.
- bool [automaticClutch](#) = true
Will clutch be automatically operated or will user input be used to operate it?
- float [clutchPedalPressedPercent](#)
0 for fully released and 1 for fully depressed pedal.
- AnimationCurve [clutchEngagementCurve](#)
Describes how much clutch will 'grab' as the pedal is released. When the Y of the curve is at 1 this means that the clutch is fully engaged, i.e. there is no slip between transmission and engine. When the Y of the curve is at 0 there is no connection between engine and transmission / wheels. 0 on the X axis represents fully released clutch pedal while 1 represents fully pressed pedal. Normally every clutch would have two points, [0,1] and [1,0], and the in-between will vary from vehicle to vehicle.
- float [targetClutchRPM](#) = 1500
Engine will try and hold RPM at this value while the clutch is being released.
- float [lastShiftTime](#)
Time since the start of the scene when the last shift happened.

Properties

- float [ClutchPercent](#) [get]
Indicates the state the clutch is in. 0 - clutch is released and engine is connected to the wheels. 1 - clutch is pressed and engine is free revving.
- float [AddedClutchRPM](#) [get]
- int [ForwardGearCount](#) [get]
Number of forward gears.
- int [ReverseGearCount](#) [get]

- Number of reverse gears.*

 - List< float > **ForwardGears** [get, set]

List of forward gears. Gears list will be updated if new value is assigned.
- List< float > **ReverseGears** [get, set]

List of reverse gears. Gears list will be updated if new value is assigned.
- float **AdjustedShiftUpRpm** [get]

Engine RPM at which transmission will shift up if dynamic shift point is enabled.
- float **AdjustedShiftDownRpm** [get]

Engine RPM at which transmission will shift down if dynamic shift point is enabled.
- int **Gear** [get, set]

0 for neutral, less than 0 for reverse gears and lager than 0 for forward gears.
- string **GearName** [get]

Returns current gear name as a string, e.g. "R", "R2", "N" or "1"
- List< float > **Gears** [get]

List of all gear ratios including reverse, forward and neutral gears. e.g. -2nd, -1st, 0 (netural), 1st, 2nd, 3rd, etc.
- float **GearRatio** [get]

Total gear ratio of the transmission for current gear.
- float **RPM** [get]

RPM at the axle side.
- float **ReverseRPM** [get]

RPM at the engine side calculated from the RPM at the axle side and gear ratios.
- bool **CanShift** [get]

True if shifting is allowed at the moment.
- bool **Shifting** [get]

True if currently shifting.

6.63.1 Detailed Description

Handles gear changing and also torque and RPM transmission in both directions.

6.63.2 Member Enumeration Documentation

6.63.2.1 DifferentialType

```
enum NWH.VehiclePhysics.Transmission.DifferentialType [strong]
```

Enumerator

Equal	
Open	
LimitedSlip	
Locking	

6.63.2.2 TransmissionType

```
enum NWH.VehiclePhysics.Transmission.TransmissionType [strong]
```

Enumerator

Manual	
Automatic	
AutomaticSequential	

6.63.3 Member Function Documentation

6.63.3.1 GetClutchEngagementAtPedalPosition()

```
float NWH.VehiclePhysics.Transmission.GetClutchEngagementAtPedalPosition (
    float clutchPercent )
```

Returns clutch value in range of 1 (disconnected) to 0 (connected) for the passed pedal travel.

6.63.3.2 GetGearRatio()

```
float NWH.VehiclePhysics.Transmission.GetGearRatio (
    int g )
```

Total gear ratio of the transmission for the specific gear.

Returns

6.63.3.3 GetMaxSpeedForGear()

```
float NWH.VehiclePhysics.Transmission.GetMaxSpeedForGear (
    int g )
```

Returns maximum speed for the engine's maxRPM and gear ratio.

6.63.3.4 Initialize()

```
void NWH.VehiclePhysics.Transmission.Initialize (
    VehicleController vc )
```

6.63.3.5 ReconstructGearList()

```
void NWH.VehiclePhysics.Transmission.ReconstructGearList ( )
```

Recreates gear list from the forward and reverse gears lists.

6.63.3.6 ReverseTransmitRPM() [1/2]

```
float NWH.VehiclePhysics.Transmission.ReverseTransmitRPM (
    float inputRPM )
```

Converts axle RPM to engine RPM.

6.63.3.7 ReverseTransmitRPM() [2/2]

```
float NWH.VehiclePhysics.Transmission.ReverseTransmitRPM (
    float inputRPM,
    int g )
```

Converts axle RPM to engine RPM for given gear in Gears list.

6.63.3.8 ReverseTransmitTorque()

```
float NWH.VehiclePhysics.Transmission.ReverseTransmitTorque (
    float inputTorque )
```

Converts axle torque to engine torque.

6.63.3.9 ShiftInto()

```
void NWH.VehiclePhysics.Transmission.ShiftInto (
    int g )
```

Shifts into given gear. 0 for neutral, less than 0 for reverse and above 0 for forward gears.

6.63.3.10 TorqueSplit()

```
void NWH.VehiclePhysics.Transmission.TorqueSplit (
    float torque,
    float topRPM )
```

Distribute torque between axles.

Parameters

<i>torque</i>	Input torque
<i>topRPM</i>	Input RPM

6.63.3.11 TransmitRPM()

```
float NWH.VehiclePhysics.Transmission.TransmitRPM (
    float inputRPM )
```

Converts engine RPM to axle RPM.

6.63.3.12 TransmitTorque()

```
float NWH.VehiclePhysics.Transmission.TransmitTorque (
    float inputTorque )
```

Converts engine torque to axle torque.

6.63.3.13 Update()

```
void NWH.VehiclePhysics.Transmission.Update ( )
```

6.63.3.14 UpdateClutch()

```
void NWH.VehiclePhysics.Transmission.UpdateClutch ( )
```

Updates clutch state.

6.63.4 Member Data Documentation**6.63.4.1 automaticClutch**

```
bool NWH.VehiclePhysics.Transmission.automaticClutch = true
```

Will clutch be automatically operated or will user input be used to operate it?

6.63.4.2 clutchEngagementCurve

AnimationCurve NWH.VehiclePhysics.Transmission.clutchEngagementCurve

Initial value:

```
= new AnimationCurve(new Keyframe[2] {  
    new Keyframe(0f, 1f),  
    new Keyframe(1f, 0f)  
})
```

Describes how much clutch will 'grab' as the pedal is released. When the Y of the curve is at 1 this means that the clutch is fully engaged, i.e. there is no slip between transmission and engine. When the Y of the curve is at 0 there is no connection between engine and transmission / wheels. 0 on the X axis represents fully released clutch pedal while 1 represents fully pressed pedal. Normally every clutch would have two points, [0,1] and [1,0], and the in-between will vary from vehicle to vehicle.

6.63.4.3 clutchPedalPressedPercent

float NWH.VehiclePhysics.Transmission.clutchPedalPressedPercent

0 for fully released and 1 for fully depressed pedal.

6.63.4.4 differentialType

DifferentialType NWH.VehiclePhysics.Transmission.differentialType = DifferentialType.Equal

Differential(s) that determine how the torque will be split between axles. Each axle then has its own differential (check axle settings).

6.63.4.5 dynamicShiftPoint

bool NWH.VehiclePhysics.Transmission.dynamicShiftPoint = true

If enabled transmission will adjust both shift up and down points to match current load.

6.63.4.6 gearMultiplier

float NWH.VehiclePhysics.Transmission.gearMultiplier = 1

Final gear multiplier. Each gear gets multiplied by this value.

6.63.4.7 lastShiftTime

```
float NWH.VehiclePhysics.Transmission.lastShiftTime
```

Time since the start of the scene when the last shift happened.

6.63.4.8 postShiftBan

```
float NWH.VehiclePhysics.Transmission.postShiftBan = 0.5f
```

Time after shifting in which shifting can not be done again.

6.63.4.9 shiftDuration

```
float NWH.VehiclePhysics.Transmission.shiftDuration = 0.2f
```

Time it takes transmission to shift between gears.

6.63.4.10 shiftDurationRandomness

```
float NWH.VehiclePhysics.Transmission.shiftDurationRandomness = 0.2f
```

Maximum percentage of shift duration that will be added or subtracted to it. Default is 20% (0.2f).

6.63.4.11 shiftPointRandomness

```
float NWH.VehiclePhysics.Transmission.shiftPointRandomness = 0.05f
```

Shift point will randomly vary by the following percent of it's value.

6.63.4.12 targetClutchRPM

```
float NWH.VehiclePhysics.Transmission.targetClutchRPM = 1500
```

Engine will try and hold RPM at this value while the clutch is being released.

6.63.4.13 targetShiftDownRPM

```
float NWH.VehiclePhysics.Transmission.targetShiftDownRPM = 1400
```

RPM at which automatic transmission will shift down. If dynamic shift point is enabled this value will change depending on load.

6.63.4.14 targetShiftUpRPM

```
float NWH.VehiclePhysics.Transmission.targetShiftUpRPM = 3600
```

RPM at which automatic transmission will shift up. If dynamic shift point is enabled this value will change depending on load.

6.63.4.15 transmissionType

```
TransmissionType NWH.VehiclePhysics.Transmission.transmissionType = TransmissionTypeAutomaticSequential
```

Determines in which way gears can be changed. Manual - gears can only be shifted by manual user input. Automatic - automatic gear changing. Allows for gear skipping (e.g. 3rd->5th) which can be useful in trucks and other high gear count vehicles. AutomaticSequential - automatic gear changing but only one gear at the time can be shifted (e.g. 3rd->4th)

6.63.5 Property Documentation

6.63.5.1 AddedClutchRPM

```
float NWH.VehiclePhysics.Transmission.AddedClutchRPM [get]
```

6.63.5.2 AdjustedShiftDownRpm

```
float NWH.VehiclePhysics.Transmission.AdjustedShiftDownRpm [get]
```

[Engine](#) RPM at which transmission will shift down if dynamic shift point is enabled.

6.63.5.3 AdjustedShiftUpRpm

```
float NWH.VehiclePhysics.Transmission.AdjustedShiftUpRpm [get]
```

Engine RPM at which transmission will shift up if dynamic shift point is enabled.

6.63.5.4 CanShift

```
bool NWH.VehiclePhysics.Transmission.CanShift [get]
```

True if shifting is allowed at the moment.

6.63.5.5 ClutchPercent

```
float NWH.VehiclePhysics.Transmission.ClutchPercent [get]
```

Indicates the state the clutch is in. 0 - clutch is released and engine is connected to the wheels. 1 - clutch is pressed and engine is free revving.

6.63.5.6 ForwardGearCount

```
int NWH.VehiclePhysics.Transmission.ForwardGearCount [get]
```

Number of forward gears.

6.63.5.7 ForwardGears

```
List<float> NWH.VehiclePhysics.Transmission.ForwardGears [get], [set]
```

List of forward gears. Gears list will be updated if new value is assigned.

6.63.5.8 Gear

```
int NWH.VehiclePhysics.Transmission.Gear [get], [set]
```

0 for neutral, less than 0 for reverse gears and larger than 0 for forward gears.

6.63.5.9 GearName

```
string NWH.VehiclePhysics.Transmission.GearName [get]
```

Returns current gear name as a string, e.g. "R", "R2", "N" or "1"

6.63.5.10 GearRatio

```
float NWH.VehiclePhysics.Transmission.GearRatio [get]
```

Total gear ratio of the transmission for current gear.

6.63.5.11 Gears

```
List<float> NWH.VehiclePhysics.Transmission.Gears [get]
```

List of all gear ratios including reverse, forward and neutral gears. e.g. -2nd, -1st, 0 (netural), 1st, 2nd, 3rd, etc.

6.63.5.12 ReverseGearCount

```
int NWH.VehiclePhysics.Transmission.ReverseGearCount [get]
```

Number of reverse gears.

6.63.5.13 ReverseGears

```
List<float> NWH.VehiclePhysics.Transmission.ReverseGears [get], [set]
```

List of reverse gears. Gears list will be updated if new value is assigned.

6.63.5.14 ReverseRPM

```
float NWH.VehiclePhysics.Transmission.ReverseRPM [get]
```

RPM at the engine side calculated from the RPM at the axle side and gear ratios.

6.63.5.15 RPM

```
float NWH.VehiclePhysics.Transmission.RPM [get]
```

RPM at the axle side.

6.63.5.16 Shifting

```
bool NWH.VehiclePhysics.Transmission.Shifting [get]
```

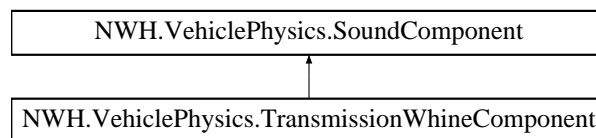
True if currently shifting.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Transmission.cs](#)

6.64 NWH.VehiclePhysics.TransmissionWhineComponent Class Reference

Inheritance diagram for NWH.VehiclePhysics.TransmissionWhineComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController vc](#), AudioManagerGroup amg)
- override void [Update](#) ()

Public Attributes

- float [pitchRange](#) = 0.2f
Pitch range that will be added to the base pitch depending on transmission state.

Additional Inherited Members

6.64.1 Member Function Documentation

6.64.1.1 Initialize()

```
override void NWH.VehiclePhysics.TransmissionWhineComponent.Initialize (
    VehicleController vc,
    AudioMixerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.64.1.2 Update()

```
override void NWH.VehiclePhysics.TransmissionWhineComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.64.2 Member Data Documentation

6.64.2.1 pitchRange

```
float NWH.VehiclePhysics.TransmissionWhineComponent.pitchRange = 0.2f
```

Pitch range that will be added to the base pitch depending on transmission state.

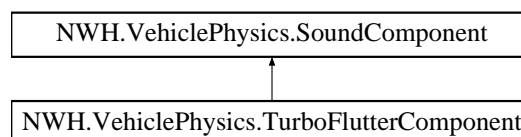
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[TransmissionWhineComponent.cs](#)

6.65 NWH.VehiclePhysics.TurboFlutterComponent Class Reference

[Sound](#) of wastegate releasing air on turbocharged vehicles.

Inheritance diagram for NWH.VehiclePhysics.TurboFlutterComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, AudioMixerGroup amg)
- override void [Update](#) ()

Additional Inherited Members

6.65.1 Detailed Description

[Sound](#) of wastegate releasing air on turbocharged vehicles.

6.65.2 Member Function Documentation

6.65.2.1 Initialize()

```
override void NWH.VehiclePhysics.TurboFlutterComponent.Initialize (
    VehicleController vc,
    AudioMixerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.65.2.2 Update()

```
override void NWH.VehiclePhysics.TurboFlutterComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

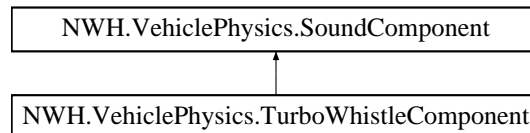
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[TurboFlutterComponent.cs](#)

6.66 NWH.VehiclePhysics.TurboWhistleComponent Class Reference

[Sound](#) of turbocharger or supercharger.

Inheritance diagram for NWH.VehiclePhysics.TurboWhistleComponent:



Public Member Functions

- override void [Initialize](#) ([VehicleController](#) vc, AudioMixerGroup amg)
- override void [Update](#) ()

Public Attributes

- float `pitchRange` = 1.4f
Pitch range that will be added to the base pitch depending on turbos's RPM.

Additional Inherited Members

6.66.1 Detailed Description

[Sound](#) of turbocharger or supercharger.

6.66.2 Member Function Documentation

6.66.2.1 Initialize()

```
override void NWH.VehiclePhysics.TurboWhistleComponent.Initialize (
    VehicleController vc,
    AudioManagerGroup amg ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.66.2.2 Update()

```
override void NWH.VehiclePhysics.TurboWhistleComponent.Update ( ) [virtual]
```

Implements [NWH.VehiclePhysics.SoundComponent](#).

6.66.3 Member Data Documentation

6.66.3.1 pitchRange

```
float NWH.VehiclePhysics.TurboWhistleComponent.pitchRange = 1.4f
```

Pitch range that will be added to the base pitch depending on turbos's RPM.

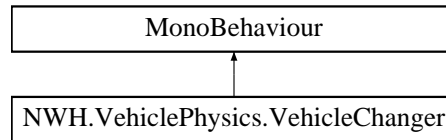
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/[TurboWhistleComponent.cs](#)

6.67 NWH.VehiclePhysics.VehicleChanger Class Reference

Used for changing vehicles. Also activates and deactivates vehicle cameras if default VehicleCamera system is used.

Inheritance diagram for NWH.VehiclePhysics.VehicleChanger:



Public Member Functions

- [VehicleController NearestVehicleFrom](#) (GameObject go)
Finds nearest vehicle on the vehicles list.
- void [NextVehicle](#) ()
Changes vehicle to a next vehicle on the Vehicles list.
- void [ChangeVehicle](#) ([VehicleController](#) vc)
Changes vehicle to a vehicle with the requested name if there is such a vehicle.
- void [ChangeVehicle](#) (int index)
Changes vehicle to requested vehicle.

Public Attributes

- List< [VehicleController](#) > [vehicles](#) = new List<[VehicleController](#)>()
List of all of the vehicles that can be selected and driven in the scene.
- string [vehicleTag](#) = "Vehicle"
Tag that the script will search for if vehicles list is empty. Can be left empty if vehicles have already been assigned manually.

Properties

- [VehicleController CurrentVehicle](#) [get]
Returns current selected vehicle.
- [VehicleController ActiveVehicleController](#) [get]
Returns currently active vehicle controller or null if none are active.
- bool [DeactivateAll](#) [get, set]
- bool [CharacterBased](#) [get, set]

6.67.1 Detailed Description

Used for changing vehicles. Also activates and deactivates vehicle cameras if default VehicleCamera system is used.

6.67.2 Member Function Documentation

6.67.2.1 ChangeVehicle() [1/2]

```
void NWH.VehiclePhysics.VehicleChanger.ChangeVehicle (
    VehicleController vc )
```

Changes vehicle to a vehicle with the requested name if there is such a vehicle.

6.67.2.2 ChangeVehicle() [2/2]

```
void NWH.VehiclePhysics.VehicleChanger.ChangeVehicle (
    int index )
```

Changes vehicle to requested vehicle.

Parameters

<i>index</i>	Index of a vehicle in Vehicles list.
--------------	--------------------------------------

6.67.2.3 NearestVehicleFrom()

```
VehicleController NWH.VehiclePhysics.VehicleChanger.NearestVehicleFrom (
    GameObject go )
```

Finds nearest vehicle on the vehicles list.

6.67.2.4 NextVehicle()

```
void NWH.VehiclePhysics.VehicleChanger.NextVehicle ( )
```

Changes vehicle to a next vehicle on the Vehicles list.

6.67.3 Member Data Documentation

6.67.3.1 vehicles

```
List<VehicleController> NWH.VehiclePhysics.VehicleChanger.vehicles = new List<VehicleController>()
```

List of all of the vehicles that can be selected and driven in the scene.

6.67.3.2 vehicleTag

```
string NWH.VehiclePhysics.VehicleChanger.vehicleTag = "Vehicle"
```

Tag that the script will search for if vehicles list is empty. Can be left empty if vehicles have already been assigned manually.

6.67.4 Property Documentation

6.67.4.1 ActiveVehicleController

```
VehicleController NWH.VehiclePhysics.VehicleChanger.ActiveVehicleController [get]
```

Returns currently active vehicle controller or null if none are active.

6.67.4.2 CharacterBased

```
bool NWH.VehiclePhysics.VehicleChanger.CharacterBased [get], [set]
```

6.67.4.3 CurrentVehicle

```
VehicleController NWH.VehiclePhysics.VehicleChanger.CurrentVehicle [get]
```

Returns current selected vehicle.

6.67.4.4 DeactivateAll

```
bool NWH.VehiclePhysics.VehicleChanger.DeactivateAll [get], [set]
```

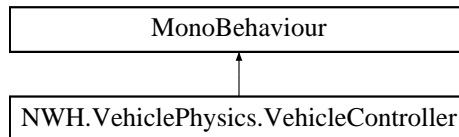
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/[VehicleChanger.cs](#)

6.68 NWH.VehiclePhysics.VehicleController Class Reference

Main class controlling all the other parts of the vehicle.

Inheritance diagram for NWH.VehiclePhysics.VehicleController:



Public Types

- enum [VehicleCollisionState](#) { [VehicleCollisionState.Enter](#), [VehicleCollisionState.Stay](#), [VehicleCollisionState.None](#) }

Public Member Functions

- void [Update](#) ()
- void [LateUpdate](#) ()
- float [GetAverageWheelRPM](#) ()
Average RPM of all wheels on the vehicle.
- float [GetRawAverageWheelRPM](#) ()
Average unsmoothed RPM of all wheels on the vehicle.
- float [GetCorrectWheelRpm](#) ([Wheel](#) wheel)
- bool [DetectWheelSpin](#) ()
True if any of the wheels is spinning.
- bool [DetectWheelSkid](#) ()
True if any of the wheels is drifting / skidding.
- [VehicleCollisionState](#) [GetCollisionState](#) ()
- void [Reset](#) ()
- void [SetDefaults](#) ()
Resets the vehicle to default state. Will find WheelControllers, assign axles, set default values for all fields and assign default audio clips from resources folder.
- void [ResetCOM](#) ()
Finds and resets center of mass of the vehicle in relation to the position of all WheelController components.
- List< [Axle](#) > [GetAxles](#) ()
Gets all the axles the vehicle has. Will search all child objects for WheelController components and return a list of axles, each with 2 Wheel Controllers paired by their Z position relative to the parent. Axles will be returned in order from front to back.

Static Public Member Functions

- static float [AngleSigned](#) (Vector3 v1, Vector3 v2, Vector3 n)
Calculates an angle between two vectors in relation a normal.
- static float [GetMouseHorizontal](#) ()
- static float [GetMouseVertical](#) ()

Public Attributes

- [InputStates](#) `input` = new [InputStates](#)()
- [Sound](#) `sound` = new [Sound](#)()
- [Effects](#) `effects` = new [Effects](#)()
- [Steering](#) `steering` = new [Steering](#)()
- [Engine](#) `engine` = new [Engine](#)()
- [Transmission](#) `transmission` = new [Transmission](#)()
- `List< Axle > axles` = new `List< Axle>()`
- [Brakes](#) `brakes` = new [Brakes](#)()
- [Tracks](#) `tracks` = new [Tracks](#)()
- [GroundDetection](#) `groundDetection`
- [DrivingAssists](#) `drivingAids` = new [DrivingAssists](#)()
- [DamageHandler](#) `damage` = new [DamageHandler](#)()
- [Fuel](#) `fuel` = new [Fuel](#)()
- [Rigging](#) `rigging` = new [Rigging](#)()
- [FlipOver](#) `flipOver` = new [FlipOver](#)()
- [TrailerHandler](#) `trailer` = new [TrailerHandler](#)()
- [Metrics](#) `metrics` = new [Metrics](#)()
- float `forwardSlipThreshold` = 0.35f
- float `sideSlipThreshold` = 0.1f
- float `speedLimiter` = 0f
- Rigidbody `vehicleRigidbody`

Properties

- bool [Active](#) [get, set]
If true vehicle can be driven. If false vehicle will be in a suspended state with only the minimal functions working and will not respond to input.
- float [ForwardVelocity](#) [get]
Velocity in forward direction in local coordinates (z-forward).
- float [Speed](#) [get]
Speed in forward direction in local coordinates (z-forward). Always positive. For positive/negative version use ForwardVelocity.
- float [SpeedKPH](#) [get]
Speed in kilometers per hour.
- float [SpeedMPH](#) [get]
Speed is (US) miles per hour.
- float [Load](#) [get]
Amount of load vehicle / engine is under. 0 to 1.
- float [WheelSpeed](#) [get]
Speed at the wheels. Only powered wheels are taken into account. If no powered axles ForwardVelocity of rigidbody will be returned instead.
- float [ForwardAcceleration](#) [get]
Acceleration in forward direction in local coordinates (z-forward).
- Vector3 [Acceleration](#) [get]
Acceleration in local coordinates (z-forward)
- float [Direction](#) [get]
Direction the vehicle is currently traveling in. 1 for forward, -1 for reverse and 0 for being perfectly still.
- List< [Wheel](#) > [Wheels](#) [get]
List of all wheels attached to this vehicle.
- bool [WheelSpin](#) [get]

- True if any of the wheels is spinning out (slipping in the forward direction).*
 • bool [WheelSkid](#) [get]
- True if any of the wheels is skidding (slipping in the lateral / side direction)*
 • [VehicleCollisionState](#) [CollisionState](#) [get]
- Returns the state of the current collision.*
 • Collision [CollisionInfo](#) [get]
- Returns the info on the current collision.*

6.68.1 Detailed Description

Main class controlling all the other parts of the vehicle.

6.68.2 Member Enumeration Documentation

6.68.2.1 VehicleCollisionState

```
enum NWH.VehiclePhysics.VehicleController.VehicleCollisionState [strong]
```

Enumerator

Enter	
Stay	
None	

6.68.3 Member Function Documentation

6.68.3.1 AngleSigned()

```
static float NWH.VehiclePhysics.VehicleController.AngleSigned (
    Vector3 v1,
    Vector3 v2,
    Vector3 n ) [static]
```

Calculates an angle between two vectors in relation a normal.

Parameters

<i>v1</i>	First Vector.
<i>v2</i>	Second Vector.
<i>n</i>	Angle around this vector.

Returns

Angle in degrees.

6.68.3.2 DetectWheelSkid()

```
bool NWH.VehiclePhysics.VehicleController.DetectWheelSkid ( )
```

True if any of the wheels is drifting / skidding.

6.68.3.3 DetectWheelSpin()

```
bool NWH.VehiclePhysics.VehicleController.DetectWheelSpin ( )
```

True if any of the wheels is spinning.

Returns**6.68.3.4 GetAverageWheelRPM()**

```
float NWH.VehiclePhysics.VehicleController.GetAverageWheelRPM ( )
```

Average RPM of all wheels on the vehicle.

6.68.3.5 GetAxles()

```
List<Axle> NWH.VehiclePhysics.VehicleController.GetAxles ( )
```

Gets all the axles the vehicle has. Will search all child objects for WheelController components and return a list of axles, each with 2 Wheel Controllers paired by their Z position relative to the parent. Axles will be returned in order from front to back.

6.68.3.6 GetCollisionState()

```
VehicleCollisionState NWH.VehiclePhysics.VehicleController.GetCollisionState ( )
```

6.68.3.7 GetCorrectWheelRpm()

```
float NWH.VehiclePhysics.VehicleController.GetCorrectWheelRpm (
    Wheel wheel )
```

6.68.3.8 GetMouseHorizontal()

```
static float NWH.VehiclePhysics.VehicleController.GetMouseHorizontal ( ) [static]
```

6.68.3.9 GetMouseVertical()

```
static float NWH.VehiclePhysics.VehicleController.GetMouseVertical ( ) [static]
```

6.68.3.10 GetRawAverageWheelRPM()

```
float NWH.VehiclePhysics.VehicleController.GetRawAverageWheelRPM ( )
```

Average unsmoothed RPM of all wheels on the vehicle.

Returns

6.68.3.11 LateUpdate()

```
void NWH.VehiclePhysics.VehicleController.LateUpdate ( )
```

6.68.3.12 Reset()

```
void NWH.VehiclePhysics.VehicleController.Reset ( )
```

6.68.3.13 ResetCOM()

```
void NWH.VehiclePhysics.VehicleController.ResetCOM ( )
```

Finds and resets center of mass of the vehicle in relation to the position of all WheelController components.

6.68.3.14 SetDefaults()

```
void NWH.VehiclePhysics.VehicleController.SetDefaults ( )
```

Resets the vehicle to default state. Will find WheelControllers, assign axles, set default values for all fields and assign default audio clips from resources folder.

6.68.3.15 Update()

```
void NWH.VehiclePhysics.VehicleController.Update ( )
```

6.68.4 Member Data Documentation

6.68.4.1 axles

```
List<Axle> NWH.VehiclePhysics.VehicleController.axles = new List<Axle>()
```

6.68.4.2 brakes

```
Brakes NWH.VehiclePhysics.VehicleController.brakes = new Brakes()
```

6.68.4.3 damage

```
DamageHandler NWH.VehiclePhysics.VehicleController.damage = new DamageHandler()
```

6.68.4.4 drivingAids

```
DrivingAssists NWH.VehiclePhysics.VehicleController.drivingAids = new DrivingAssists()
```

6.68.4.5 effects

```
Effects NWH.VehiclePhysics.VehicleController.effects = new Effects()
```

6.68.4.6 engine

```
Engine NWH.VehiclePhysics.VehicleController.engine = new Engine()
```

6.68.4.7 flipOver

```
FlipOver NWH.VehiclePhysics.VehicleController.flipOver = new FlipOver()
```

6.68.4.8 forwardSlipThreshold

```
float NWH.VehiclePhysics.VehicleController.forwardSlipThreshold = 0.35f
```

6.68.4.9 fuel

```
Fuel NWH.VehiclePhysics.VehicleController.fuel = new Fuel()
```

6.68.4.10 groundDetection

```
GroundDetection NWH.VehiclePhysics.VehicleController.groundDetection
```

6.68.4.11 input

```
InputStates NWH.VehiclePhysics.VehicleController.input = new InputStates()
```


6.68.4.12 metrics

```
Metrics NWH.VehiclePhysics.VehicleController.metrics = new Metrics()
```

6.68.4.13 rigging

```
Rigging NWH.VehiclePhysics.VehicleController.rigging = new Rigging()
```

6.68.4.14 sideSlipThreshold

```
float NWH.VehiclePhysics.VehicleController.sideSlipThreshold = 0.1f
```

6.68.4.15 sound

```
Sound NWH.VehiclePhysics.VehicleController.sound = new Sound()
```

6.68.4.16 speedLimiter

```
float NWH.VehiclePhysics.VehicleController.speedLimiter = 0f
```

6.68.4.17 steering

```
Steering NWH.VehiclePhysics.VehicleController.steering = new Steering()
```

6.68.4.18 tracks

```
Tracks NWH.VehiclePhysics.VehicleController.tracks = new Tracks()
```

6.68.4.19 trailer

```
TrailerHandler NWH.VehiclePhysics.VehicleController.trailer = new TrailerHandler()
```

6.68.4.20 transmission

```
Transmission NWH.VehiclePhysics.VehicleController.transmission = new Transmission()
```

6.68.4.21 vehicleRigidbody

```
Rigidbody NWH.VehiclePhysics.VehicleController.vehicleRigidbody
```

6.68.5 Property Documentation

6.68.5.1 Acceleration

```
Vector3 NWH.VehiclePhysics.VehicleController.Acceleration [get]
```

Acceleration in local coordinates (z-forward)

6.68.5.2 Active

```
bool NWH.VehiclePhysics.VehicleController.Active [get], [set]
```

If true vehicle can be driven. If false vehicle will be in a suspended state with only the minimal functions working and will not respond to input.

6.68.5.3 CollisionInfo

```
Collision NWH.VehiclePhysics.VehicleController.CollisionInfo [get]
```

Returns the info on the current collision.

6.68.5.4 CollisionState

```
VehicleCollisionState NWH.VehiclePhysics.VehicleController.CollisionState [get]
```

Returns the state of the current collision.

6.68.5.5 Direction

```
float NWH.VehiclePhysics.VehicleController.Direction [get]
```

Direction the vehicle is currently traveling in. 1 for forward, -1 for reverse and 0 for being perfectly still.

6.68.5.6 ForwardAcceleration

```
float NWH.VehiclePhysics.VehicleController.ForwardAcceleration [get]
```

Acceleration in forward direction in local coordinates (z-forward).

6.68.5.7 ForwardVelocity

```
float NWH.VehiclePhysics.VehicleController.ForwardVelocity [get]
```

Velocity in forward direction in local coordinates (z-forward).

6.68.5.8 Load

```
float NWH.VehiclePhysics.VehicleController.Load [get]
```

Amount of load vehicle / engine is under. 0 to 1.

6.68.5.9 Speed

```
float NWH.VehiclePhysics.VehicleController.Speed [get]
```

Speed in forward direction in local coordinates (z-forward). Always positive. For positive/negative version use ForwardVelocity.

6.68.5.10 SpeedKPH

```
float NWH.VehiclePhysics.VehicleController.SpeedKPH [get]
```

Speed in kilometers per hour.

6.68.5.11 SpeedMPH

```
float NWH.VehiclePhysics.VehicleController.SpeedMPH [get]
```

Speed is (US) miles per hour.

6.68.5.12 Wheels

```
List<Wheel> NWH.VehiclePhysics.VehicleController.Wheels [get]
```

List of all wheels attached to this vehicle.

6.68.5.13 WheelSkid

```
bool NWH.VehiclePhysics.VehicleController.WheelSkid [get]
```

True if any of the wheels is skidding (slipping in the lateral / side direction)

6.68.5.14 WheelSpeed

```
float NWH.VehiclePhysics.VehicleController.WheelSpeed [get]
```

Speed at the wheels. Only powered wheels are taken into account. If no powered axles ForwardVelocity of rigidbody will be returned instead.

6.68.5.15 WheelSpin

```
bool NWH.VehiclePhysics.VehicleController.WheelSpin [get]
```

True if any of the wheels is spinning out (slipping in the forward direction).

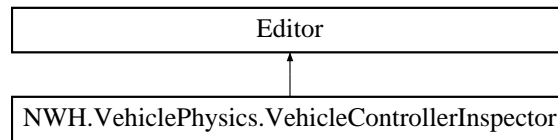
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[VehicleController.cs](#)

6.69 NWH.VehiclePhysics.VehicleControllerInspector Class Reference

Shows different settings for trailer depending if vehicle is a trailer or a towing vehicle.

Inheritance diagram for NWH.VehiclePhysics.VehicleControllerInspector:



Public Member Functions

- override void [OnInspectorGUI](#) ()

6.69.1 Detailed Description

Shows different settings for trailer depending if vehicle is a trailer or a towing vehicle.

6.69.2 Member Function Documentation

6.69.2.1 OnInspectorGUI()

```
override void NWH.VehiclePhysics.VehicleControllerInspector.OnInspectorGUI ( )
```

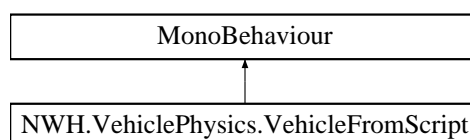
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Editor/[VehicleControllerInspector.cs](#)

6.70 NWH.VehiclePhysics.VehicleFromScript Class Reference

Example script for setting up a vehicle at runtime - modify as/if needed. Sets up a vehicle from script at runtime. Model is required to already have body colliders set up, as well as wheels tagged with correct tag. Model also needs to have correct rotation (Z-forward, Y-up, X-right). Works only on vehicles with two wheels per axle. Number of axles is not limited.

Inheritance diagram for NWH.VehiclePhysics.VehicleFromScript:



Public Attributes

- string `wheelTag` = "Wheel"
- string `ignoreLayer` = "WheelControllerIgnore"
- float `vehicleMass` = 1300
- float `wheelRadius` = 0.3f
- float `wheelWidth` = 0.28f

6.70.1 Detailed Description

Example script for setting up a vehicle at runtime - modify as/if needed. Sets up a vehicle from script at runtime. Model is required to already have body colliders set up, as well as wheels tagged with correct tag. Model also needs to have correct rotation (Z-forward, Y-up, X-right). Works only on vehicles with two wheels per axle. Number of axles is not limited.

6.70.2 Member Data Documentation

6.70.2.1 ignoreLayer

```
string NWH.VehiclePhysics.VehicleFromScript.ignoreLayer = "WheelControllerIgnore"
```

6.70.2.2 vehicleMass

```
float NWH.VehiclePhysics.VehicleFromScript.vehicleMass = 1300
```

6.70.2.3 wheelRadius

```
float NWH.VehiclePhysics.VehicleFromScript.wheelRadius = 0.3f
```

6.70.2.4 wheelTag

```
string NWH.VehiclePhysics.VehicleFromScript.wheelTag = "Wheel"
```

6.70.2.5 wheelWidth

```
float NWH.VehiclePhysics.VehicleFromScript.wheelWidth = 0.28f
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/[VehicleFromScript.cs](#)

6.71 NWH.VehiclePhysics.Lights.VehicleLight Class Reference

Single vehicle light.

Public Member Functions

- void [TurnOn](#) ()
Turns on the light source or enables emission on the mesh. Mesh is required to have standard shader.
- void [TurnOff](#) ()
Turns off the light source or disables emission on the mesh. Mesh is required to have standard shader.

Public Attributes

- List< Light > [lightSources](#) = new List<Light>()
List of light sources of any type.
- List< MeshRenderer > [lightMeshRenderers](#) = new List<MeshRenderer>()
List of mesh renderers with standard shader. Emission will be turned on or off depending on light state.

Protected Attributes

- bool [active](#)

Properties

- bool [On](#) [get]
State of the light.

6.71.1 Detailed Description

Single vehicle light.

6.71.2 Member Function Documentation

6.71.2.1 TurnOff()

```
void NWH.VehiclePhysics.Lights.VehicleLight.TurnOff ( )
```

Turns off the light source or disables emission on the mesh. Mesh is required to have standard shader.

6.71.2.2 TurnOn()

```
void NWH.VehiclePhysics.Lights.VehicleLight.TurnOn ( )
```

Turns on the light source or enables emission on the mesh. Mesh is required to have standard shader.

6.71.3 Member Data Documentation

6.71.3.1 active

```
bool NWH.VehiclePhysics.Lights.VehicleLight.active [protected]
```

6.71.3.2 lightMeshRenderers

```
List<MeshRenderer> NWH.VehiclePhysics.Lights.VehicleLight.lightMeshRenderers = new List<Mesh↵  
Renderer> ( )
```

List of mesh renderers with standard shader. Emission will be turned on or off depending on light state.

6.71.3.3 lightSources

```
List<Light> NWH.VehiclePhysics.Lights.VehicleLight.lightSources = new List<Light> ( )
```

List of light sources of any type.

6.71.4 Property Documentation

6.71.4.1 On

```
bool NWH.VehiclePhysics.Lights.VehicleLight.On [get]
```

State of the light.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/[Lights.cs](#)

6.72 NWH.VehiclePhysics.Wheel Class Reference

Contains everything related to wheels. To access WC3D's properties directly use WheelController getter. Most used variables are wrapped in getters and setter to enable eventual (but unlikely) future compatibility with default wheel collider. [Wheel](#) class is not equal to WheelController class. To access WC3D (WheelController) use WheelController getter/setter.

Public Member Functions

- [Wheel](#) ()
- [Wheel](#) (WheelController wc, VehicleController vc)
- void [Update](#) ()
- void [AddBrakeTorque](#) (float torque)
Adds brake torque to the wheel on top of the existing torque. Value is clamped to max brake torque.
- void [Lockup](#) ()
Applies very high braking torque to the wheel locking it up. Unlike other methods not limited by max brake torque.
- void [Initialize](#) (VehicleController vc)
- void [SetBrakeIntensity](#) (float percent)
Adds brake torque as a percentage in range from 0 to 1.
- void [ResetBrakes](#) (float value)
Sets brake torque to the provided value. Use 0 to remove any braking.
- void [Activate](#) ()
Activates the wheel after it has been suspended by turning off single ray mode. If the wheel is in single ray mode by default it will be left on.
- void [Suspend](#) ()
Turns on single ray mode to prevent unnecessary raycasting for inactive wheels / vehicles.

Public Attributes

- float [brakeCoefficient](#) = 1f
Amount of brake torque wheel will receive as a percentage from max brake torque.
- [WheelController](#) [wheelController](#)

Properties

- float **Bias** [get, set]
Amount of motor torque this wheel will receive as a percentage from total torque on the axle.
- float **ForwardSlip** [get]
Longitudinal slip of the wheel.
- float **SideSlip** [get]
Lateral slip of the wheel.
- float **SmoothForwardSlip** [get]
Smoothed longitudinal slip of the wheel for use in effects.
- float **SmoothSideSlip** [get]
Smoothed lateral slip of the wheel for use in effects.
- float **ForwardSlipPercent** [get]
Longitudinal slip percentage where 1 represents slip equal to forward slip threshold.
- float **SideSlipPercent** [get]
Lateral slip percentage where 1 represents slip equal to side slip threshold.
- bool **HasForwardSlip** [get]
True if longitudinal slip is larger than forward slip threshold.
- **GroundDetection.GroundEntity** **CurrentGroundEntity** [get]
Returns ground entity the wheel is currently on.
- string **CurrentGroundEntityName** [get]
Returns the name of the ground entity the wheel is currently on.
- bool **HasSideSlip** [get]
True if lateral slip is larger than side slip threshold.
- float **Damage** [get, set]
Damage that the wheel has suffered so far.
- float **RPM** [get]
RPM of the wheel. In most cases it is better to use SmoothRPM instead.
- Transform **ControllerTransform** [get]
Transform to which WheelController component is attached.
- bool **IsGrounded** [get]
True if wheel is touching any object.
- float **SpringTravel** [get]
Distance from top to bottom of spring travel.
- float **MotorTorque** [get, set]
Torque in Nm used to accelerate the wheel.
- float **BrakeTorque** [get, set]
Torque in Nm used to slow down the wheel.
- Transform **VisualTransform** [get]
Transform of the object containing mesh(es) representing the wheel.
- float **Radius** [get]
Wheel radius.
- float **Width** [get]
Wheel width.
- GameObject **ControllerGO** [get]
GameObject containing WheelController component.
- float **SteerAngle** [get, set]
Steer angle of the wheel in degrees.
- float **SmoothRPM** [get]
Smoothed RPM of the wheel. Should be used instead of the actual RPM for most calculations.
- float **NoSlipRPM** [get]

- *RPM of the wheel without slipping.*
[WheelController](#) [WheelController](#) [get]
WheelController (WC3D) of the wheel.
- float [DamageSteerDirection](#) [get]
Random steer direction of a damaged wheel. Depending on the amount of the damage vehicle has received this value will be multiplied by the steer angle making the wheel gradually point more and more in a random direction drastically worsening the handling.

6.72.1 Detailed Description

Contains everything related to wheels. To access WC3D's properties directly use WheelController getter. Most used variables are wrapped in getters and setter to enable eventual (but unlikely) future compatibility with default wheel collider. [Wheel](#) class is not equal to WheelController class. To access WC3D (WheelController) use Wheel↔Controller getter/setter.

6.72.2 Constructor & Destructor Documentation

6.72.2.1 Wheel() [1/2]

```
NWH.VehiclePhysics.Wheel.Wheel ( )
```

6.72.2.2 Wheel() [2/2]

```
NWH.VehiclePhysics.Wheel.Wheel (
    WheelController wc,
    VehicleController vc )
```

6.72.3 Member Function Documentation

6.72.3.1 Activate()

```
void NWH.VehiclePhysics.Wheel.Activate ( )
```

Activates the wheel after it has been suspended by turning off single ray mode. If the wheel is in single ray mode by default it will be left on.

6.72.3.2 AddBrakeTorque()

```
void NWH.VehiclePhysics.Wheel.AddBrakeTorque (
    float torque )
```

Adds brake torque to the wheel on top of the existing torque. Value is clamped to max brake torque.

Parameters

<i>torque</i>	Torque in Nm that will be applied to the wheel to slow it down.
---------------	---

6.72.3.3 Initialize()

```
void NWH.VehiclePhysics.Wheel.Initialize (
    VehicleController vc )
```

6.72.3.4 Lockup()

```
void NWH.VehiclePhysics.Wheel.Lockup ( )
```

Applies very high braking torque to the wheel locking it up. Unlike other methods not limited by max brake torque.

6.72.3.5 ResetBrakes()

```
void NWH.VehiclePhysics.Wheel.ResetBrakes (
    float value )
```

Sets brake torque to the provided value. Use 0 to remove any braking.

6.72.3.6 SetBrakeIntensity()

```
void NWH.VehiclePhysics.Wheel.SetBrakeIntensity (
    float percent )
```

Adds brake torque as a percentage in range from 0 to 1.

6.72.3.7 Suspend()

```
void NWH.VehiclePhysics.Wheel.Suspend ( )
```

Turns on single ray mode to prevent unnecessary raycasting for inactive wheels / vehicles.

6.72.3.8 Update()

```
void NWH.VehiclePhysics.Wheel.Update ( )
```

6.72.4 Member Data Documentation

6.72.4.1 brakeCoefficient

```
float NWH.VehiclePhysics.Wheel.brakeCoefficient = 1f
```

Amount of brake torque wheel will receive as a percentage from max brake torque.

6.72.4.2 wheelController

```
WheelController NWH.VehiclePhysics.Wheel.wheelController
```

6.72.5 Property Documentation

6.72.5.1 Bias

```
float NWH.VehiclePhysics.Wheel.Bias [get], [set]
```

Amount of motor torque this wheel will receive as a percentage from total torque on the axle.

6.72.5.2 BrakeTorque

```
float NWH.VehiclePhysics.Wheel.BrakeTorque [get], [set]
```

Torque in Nm used to slow down the wheel.

6.72.5.3 ControllerGO

```
GameObject NWH.VehiclePhysics.Wheel.ControllerGO [get]
```

GameObject containing WheelController component.

6.72.5.4 ControllerTransform

`Transform NWH.VehiclePhysics.Wheel.ControllerTransform [get]`

Transform to which WheelController component is attached.

6.72.5.5 CurrentGroundEntity

`GroundDetection.GroundEntity NWH.VehiclePhysics.Wheel.CurrentGroundEntity [get]`

Returns ground entity the wheel is currently on.

6.72.5.6 CurrentGroundEntityName

`string NWH.VehiclePhysics.Wheel.CurrentGroundEntityName [get]`

Returns the name of the ground entity the wheel is currently on.

6.72.5.7 Damage

`float NWH.VehiclePhysics.Wheel.Damage [get], [set]`

Damage that the wheel has suffered so far.

6.72.5.8 DamageSteerDirection

`float NWH.VehiclePhysics.Wheel.DamageSteerDirection [get]`

Random steer direction of a damaged wheel. Depending on the amount of the damage vehicle has received this value will be multiplied by the steer angle making the wheel gradually point more and more in a random direction drastically worsening the handling.

6.72.5.9 ForwardSlip

`float NWH.VehiclePhysics.Wheel.ForwardSlip [get]`

Longitudinal slip of the wheel.

6.72.5.10 ForwardSlipPercent

```
float NWH.VehiclePhysics.Wheel.ForwardSlipPercent [get]
```

Longitudinal slip percentage where 1 represents slip equal to forward slip threshold.

6.72.5.11 HasForwardSlip

```
bool NWH.VehiclePhysics.Wheel.HasForwardSlip [get]
```

True if longitudinal slip is larger than forward slip threshold.

6.72.5.12 HasSideSlip

```
bool NWH.VehiclePhysics.Wheel.HasSideSlip [get]
```

True if lateral slip is larger than side slip threshold.

6.72.5.13 IsGrounded

```
bool NWH.VehiclePhysics.Wheel.IsGrounded [get]
```

True if wheel is touching any object.

6.72.5.14 MotorTorque

```
float NWH.VehiclePhysics.Wheel.MotorTorque [get], [set]
```

Torque in Nm used to accelerate the wheel.

6.72.5.15 NoSlipRPM

```
float NWH.VehiclePhysics.Wheel.NoSlipRPM [get]
```

RPM of the wheel without slipping.

6.72.5.16 Radius

```
float NWH.VehiclePhysics.Wheel.Radius [get]
```

[Wheel](#) radius.

6.72.5.17 RPM

```
float NWH.VehiclePhysics.Wheel.RPM [get]
```

RPM of the wheel. In most cases it is better to use SmoothRPM instead.

6.72.5.18 SideSlip

```
float NWH.VehiclePhysics.Wheel.SideSlip [get]
```

Lateral slip of the wheel.

6.72.5.19 SideSlipPercent

```
float NWH.VehiclePhysics.Wheel.SideSlipPercent [get]
```

Lateral slip percentage where 1 represents slip equal to side slip threshold.

6.72.5.20 SmoothForwardSlip

```
float NWH.VehiclePhysics.Wheel.SmoothForwardSlip [get]
```

Smoothed longitudinal slip of the wheel for use in effects.

6.72.5.21 SmoothRPM

```
float NWH.VehiclePhysics.Wheel.SmoothRPM [get]
```

Smoothed RPM of the wheel. Should be used instead of the actual RPM for most calculations.

6.72.5.22 SmoothSideSlip

```
float NWH.VehiclePhysics.Wheel.SmoothSideSlip [get]
```

Smoothed lateral slip of the wheel for use in effects.

6.72.5.23 SpringTravel

```
float NWH.VehiclePhysics.Wheel.SpringTravel [get]
```

Distance from top to bottom of spring travel.

6.72.5.24 SteerAngle

```
float NWH.VehiclePhysics.Wheel.SteerAngle [get], [set]
```

Steer angle of the wheel in degrees.

6.72.5.25 VisualTransform

```
Transform NWH.VehiclePhysics.Wheel.VisualTransform [get]
```

Transform of the object containing mesh(es) representing the wheel.

6.72.5.26 WheelController

```
WheelController NWH.VehiclePhysics.Wheel.WheelController [get]
```

WheelController (WC3D) of the wheel.

6.72.5.27 Width

```
float NWH.VehiclePhysics.Wheel.Width [get]
```

[Wheel](#) width.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/[Wheel.cs](#)

6.73 NWH.WheelController3D.WheelController.Wheel Class Reference

Contains everything wheel related, including rim and tire.

Public Member Functions

- void [Initialize](#) ([WheelController](#) wc)
Calculation of static parameters and creation of rim collider.
- void [GenerateCamberCurve](#) (float camberAtBottom, float camberAtTop)

Public Attributes

- float [mass](#) = 25.0f
- float [rimOffset](#) = 0f
- float [tireRadius](#) = 0.4f
- float [width](#) = 0.25f
- float [rpm](#)
- Vector3 [prevWorldPosition](#)
- Vector3 [worldPosition](#)
- Vector3 [prevGroundPoint](#)
- Quaternion [worldRotation](#)
- AnimationCurve [camberCurve](#) = null
- float [camberAngle](#)
- float [inertia](#)
- float [angularVelocity](#)
- float [freeRollingAngularVelocity](#)
- float [residualAngularVelocity](#)
- float [steerAngle](#)
- float [rotationAngle](#)
- GameObject [visual](#)
- GameObject [nonRotating](#)
- GameObject [rim](#)
- Transform [rimCollider](#)
- Vector3 [up](#)
- Vector3 [inside](#)
- Vector3 [forward](#)
- Vector3 [right](#)
- Vector3 [velocity](#)
- Vector3 [prevVelocity](#)
- Vector3 [acceleration](#)
- float [tireLoad](#)
- float [motorTorque](#)
- float [brakeTorque](#)
- Vector3 [nonRotatingPostionOffset](#)

6.73.1 Detailed Description

Contains everything wheel related, including rim and tire.

6.73.2 Member Function Documentation

6.73.2.1 GenerateCamberCurve()

```
void NWH.WheelController3D.WheelController.Wheel.GenerateCamberCurve (
    float camberAtBottom,
    float camberAtTop )
```

6.73.2.2 Initialize()

```
void NWH.WheelController3D.WheelController.Wheel.Initialize (
    WheelController wc )
```

Calculation of static parameters and creation of rim collider.

6.73.3 Member Data Documentation

6.73.3.1 acceleration

```
Vector3 NWH.WheelController3D.WheelController.Wheel.acceleration
```

6.73.3.2 angularVelocity

```
float NWH.WheelController3D.WheelController.Wheel.angularVelocity
```

6.73.3.3 brakeTorque

```
float NWH.WheelController3D.WheelController.Wheel.brakeTorque
```

6.73.3.4 camberAngle

```
float NWH.WheelController3D.WheelController.Wheel.camberAngle
```

6.73.3.5 camberCurve

AnimationCurve NWH.WheelController3D.WheelController.Wheel.camberCurve = null

6.73.3.6 forward

Vector3 NWH.WheelController3D.WheelController.Wheel.forward

6.73.3.7 freeRollingAngularVelocity

float NWH.WheelController3D.WheelController.Wheel.freeRollingAngularVelocity

6.73.3.8 inertia

float NWH.WheelController3D.WheelController.Wheel.inertia

6.73.3.9 inside

Vector3 NWH.WheelController3D.WheelController.Wheel.inside

6.73.3.10 mass

float NWH.WheelController3D.WheelController.Wheel.mass = 25.0f

6.73.3.11 motorTorque

float NWH.WheelController3D.WheelController.Wheel.motorTorque

6.73.3.12 nonRotating

GameObject NWH.WheelController3D.WheelController.Wheel.nonRotating

6.73.3.13 nonRotatingPostionOffset

Vector3 NWH.WheelController3D.WheelController.Wheel.nonRotatingPostionOffset

6.73.3.14 prevGroundPoint

Vector3 NWH.WheelController3D.WheelController.Wheel.prevGroundPoint

6.73.3.15 prevVelocity

Vector3 NWH.WheelController3D.WheelController.Wheel.prevVelocity

6.73.3.16 prevWorldPosition

Vector3 NWH.WheelController3D.WheelController.Wheel.prevWorldPosition

6.73.3.17 residualAngularVelocity

float NWH.WheelController3D.WheelController.Wheel.residualAngularVelocity

6.73.3.18 right

Vector3 NWH.WheelController3D.WheelController.Wheel.right

6.73.3.19 rim

GameObject NWH.WheelController3D.WheelController.Wheel.rim

6.73.3.20 rimCollider

Transform NWH.WheelController3D.WheelController.Wheel.rimCollider

6.73.3.21 rimOffset

```
float NWH.WheelController3D.WheelController.Wheel.rimOffset = 0f
```

6.73.3.22 rotationAngle

```
float NWH.WheelController3D.WheelController.Wheel.rotationAngle
```

6.73.3.23 rpm

```
float NWH.WheelController3D.WheelController.Wheel.rpm
```

6.73.3.24 steerAngle

```
float NWH.WheelController3D.WheelController.Wheel.steerAngle
```

6.73.3.25 tireLoad

```
float NWH.WheelController3D.WheelController.Wheel.tireLoad
```

6.73.3.26 tireRadius

```
float NWH.WheelController3D.WheelController.Wheel.tireRadius = 0.4f
```

6.73.3.27 up

```
Vector3 NWH.WheelController3D.WheelController.Wheel.up
```

6.73.3.28 velocity

```
Vector3 NWH.WheelController3D.WheelController.Wheel.velocity
```

6.73.3.29 visual

`GameObject NWH.WheelController3D.WheelController.Wheel.visual`

6.73.3.30 width

`float NWH.WheelController3D.WheelController.Wheel.width = 0.25f`

6.73.3.31 worldPosition

`Vector3 NWH.WheelController3D.WheelController.Wheel.worldPosition`

6.73.3.32 worldRotation

`Quaternion NWH.WheelController3D.WheelController.Wheel.worldRotation`

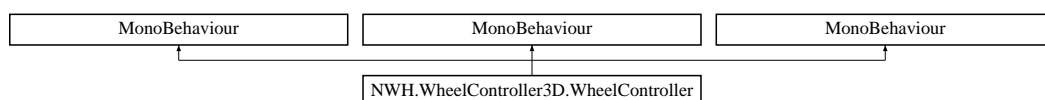
The documentation for this class was generated from the following file:

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/[WheelController.cs](#)

6.74 NWH.WheelController3D.WheelController Class Reference

API for [WheelController](#)

Inheritance diagram for NWH.WheelController3D.WheelController:



Classes

- class [Damper](#)
Suspension part.
- class [Friction](#)
All info related to longitudinal force calculation.
- class [FrictionPreset](#)
Container class for holding wheel friction presets.
- class [Spring](#)
Suspension part.
- class [Wheel](#)
Contains everything wheel related, including rim and tire.
- class [WheelHit](#)
Contains RaycastHit and extended hit data.

Public Types

- enum [Side](#) { [Side.Left](#) = -1, [Side.Right](#) = 1, [Side.Center](#) = 0, [Side.Auto](#) = 2 }
- Side of the vehicle.*

Public Member Functions

- void [GetWorldPose](#) (out Vector3 pos, out Quaternion quat)
Returns the position and rotation of the wheel.
- bool [GetGroundHit](#) (out [WheelHit](#) hit)
Returns Raycast info of the wheel's hit. Always check if the function returns true before using hit info as data will only be updated when wheel is hitting the ground (isGrounded == True).
- void [SetCamber](#) (float camberAtTop, float camberAtBottom)
Sets linear camber between the two values.
- void [SetCamber](#) (float [camber](#))
Sets fixed camber.
- void [SetCamber](#) (AnimationCurve curve)
Sets camber using AnimationCurve.
- void [SetActiveFrictionPreset](#) ([FrictionPreset](#) fp)
- void [SetActiveFrictionPreset](#) ([FrictionPreset.FrictionPresetEnum](#) fpe)
- [FrictionPreset](#) [GetFrictionPreset](#) (int index)
- void [Start](#) ()
- void [FixedUpdate](#) ()
- void [Initialize](#) ()
- Mesh [GenerateRimColliderMesh](#) (Transform rt)
- [Side](#) [DetermineSide](#) (Vector3 pointPosition, Transform referenceTransform)
Determines on what side of the vehicle a point is.

Public Attributes

- [Wheel](#) [wheel](#)
- [Spring](#) [spring](#)
- [Damper](#) [damper](#)
- [Friction](#) [fFriction](#)
Forward (longitudinal) friction info.
- [Friction](#) [sFriction](#)
Side (lateral) friction info.
- bool [debug](#)
- GameObject [parent](#)
Root object of the vehicle.
- bool [useRimCollider](#)
If enabled mesh collider mimicking the shape of rim and wheel will be positioned so that wheel can not pass through objects in case raycast does not detect the surface in time.
- [FrictionPreset.FrictionPresetEnum](#) [activeFrictionPresetEnum](#)
Current active preset enum value.
- [FrictionPreset](#) [activeFrictionPreset](#)
Current active friction preset.
- [WheelHit](#) [wheelHit](#) = new [WheelHit](#)()
Contains point in which wheel touches ground. Not valid if !isGrounded.
- bool [singleRay](#) = false
- [WheelHit](#) [singleWheelHit](#) = new [WheelHit](#)()
- bool [trackedVehicle](#) = false
Enables some wheel behaviors specific to tracked vehicles, specifically the fact that there is no wheel spin.
- bool [applyForceToOthers](#) = false
- float [maxPutDownForce](#)

Properties

- float [brakeTorque](#) [get, set]
Brake torque on the wheel axle. [Nm] Must be positive (zero included).
- bool [isGrounded](#) [get]
Is the tractive surface touching the ground? Returns false if vehicle tipped over / tire sidewall is in contact.
- float [mass](#) [get, set]
Mass of the wheel. [kg] Typical values would be in range [20, 200]
- float [motorTorque](#) [get, set]
Motor torque on the wheel axle. [Nm] Can be positive or negative based on direction.
- float [radius](#) [get, set]
Equal to tireRadius but exists because of compatibility with inbuilt WheelCollider. Radius of the complete wheel. [meters] Must be larger than 0.
- float [rimOffset](#) [get, set]
Side offset of the rim. Positive value will result if wheel further from the vehicle. [meters]
- float [tireRadius](#) [get, set]
Radius (height) of the tire. [meters]
- float [tireWidth](#) [get, set]
Width of the wheel. [meters]
- float [rpm](#) [get]
Rotations per minute of the wheel around the axle. [rpm]
- float [steerAngle](#) [get, set]
Steer angle around the wheel's up axis (with add-ons ignored). [deg]
- float [camber](#) [get]
Camber angle of the wheel. [deg] Negative angle means that the top of the wheel is closer to the vehicle than the bottom.
- float [springCompression](#) [get]
Returns value in range [0,1] where 1 means spring is fully compressed.
- float [springVelocity](#) [get]
Spring velocity in relation to local vertical axis. [m/s] Positive on rebound (extension), negative on bump (compression).
- bool [springBottomedOut](#) [get]
True when spring is fully compressed, i.e. there is no more spring travel.
- bool [springOverExtended](#) [get]
True when spring is fully extended.
- float [suspensionForce](#) [get, set]
Current spring force. [N] Can be written to for use in Anti-roll Bar script or similar.
- float [springMaximumForce](#) [get, set]
Maximum spring force. [N]
- AnimationCurve [springCurve](#) [get, set]
Spring force curve in relation to spring length.
- float [springLength](#) [get, set]
Length of the spring when fully extended.
- float [springTravel](#) [get]
Current length (travel) of spring.
- Vector3 [springTravelPoint](#) [get]
Point in which spring and swingarm are in contact.
- float [damperForce](#) [get]
Current damper force.
- float [damperUnitReboundForce](#) [get, set]
Rebounding force at 1 m/s spring velocity
- float [damperUnitBumpForce](#) [get, set]

- Bump force at 1 m/s spring velocity*
- AnimationCurve [damperCurve](#) [get, set]
Damper force curve in relation to spring velocity.
- Friction [forwardFriction](#) [get, set]
Returns `_Friction` object with longitudinal values.
- Friction [sideFriction](#) [get, set]
Returns `_Friction` object with lateral values.
- float [MaxPutDownForce](#) [get]
- Side [VehicleSide](#) [get, set]
Returns Enum [Side] with the corresponding side of the vehicle a wheel is at [Left, Right]
- float [speed](#) [get]
Returns vehicle speed in meters per second [m/s], multiply by 3.6 for [kph] or by 2.24 for [mph].
- int [ForwardScanResolution](#) [get, set]
Ground scan resolution in forward direction.
- int [SideToSideScanResolution](#) [get, set]
Number of scan planes parallel to the wheel.
- GameObject [Parent](#) [get, set]
Returns wheel's parent object.
- GameObject [Visual](#) [get, set]
Returns object that represents wheel's visual representation. Can be null in case the object is not assigned (not mandatory).
- GameObject [NonRotating](#) [get, set]
Object that follows the wheel position in everything but rotation around the axle. Can be used for brake calipers, external fenders, etc.
- Vector3 [pointVelocity](#) [get]
Returns velocity at the wheel's center position in [m/s].
- float [angularVelocity](#) [get]
Returns angular velocity of the wheel in radians. Multiply by wheel radius to get linear speed.
- LayerMask [ScanIgnoreLayers](#) [get, set]
Layers that will be ignored when doing ground detection.

6.74.1 Detailed Description

API for [WheelController](#)

6.74.2 Member Enumeration Documentation

6.74.2.1 Side

enum [NWH.WheelController3D.WheelController.Side](#) [strong]

Side of the vehicle.

Enumerator

Left	
Right	
Center	
Auto	

6.74.3 Member Function Documentation

6.74.3.1 DetermineSide()

```
Side NWH.WheelController3D.WheelController.DetermineSide (
    Vector3 pointPosition,
    Transform referenceTransform )
```

Determines on what side of the vehicle a point is.

Parameters

<i>pointPosition</i>	Position of the point in question.
<i>referenceTransform</i>	Position of the reference transform.

Returns

Enum Side [Left,Right] (int)[-1,1]

6.74.3.2 FixedUpdate()

```
void NWH.WheelController3D.WheelController.FixedUpdate ( )
```

6.74.3.3 GenerateRimColliderMesh()

```
Mesh NWH.WheelController3D.WheelController.GenerateRimColliderMesh (
    Transform rt )
```

6.74.3.4 GetFrictionPreset()

```
FrictionPreset NWH.WheelController3D.WheelController.GetFrictionPreset (
    int index )
```

6.74.3.5 GetGroundHit()

```
bool NWH.WheelController3D.WheelController.GetGroundHit (
    out WheelHit hit )
```

Returns Raycast info of the wheel's hit. Always check if the function returns true before using hit info as data will only be updated when wheel is hitting the ground (isGrounded == True).

Parameters

<i>h</i>	Standard Unity RaycastHit
----------	---------------------------

Returns**6.74.3.6 GetWorldPose()**

```
void NWH.WheelController3D.WheelController.GetWorldPose (
    out Vector3 pos,
    out Quaternion quat )
```

Returns the position and rotation of the wheel.

6.74.3.7 Initialize()

```
void NWH.WheelController3D.WheelController.Initialize ( )
```

6.74.3.8 SetActiveFrictionPreset() [1/2]

```
void NWH.WheelController3D.WheelController.SetActiveFrictionPreset (
    FrictionPreset fp )
```

6.74.3.9 SetActiveFrictionPreset() [2/2]

```
void NWH.WheelController3D.WheelController.SetActiveFrictionPreset (
    FrictionPreset.FrictionPresetEnum fpe )
```

6.74.3.10 SetCamber() [1/3]

```
void NWH.WheelController3D.WheelController.SetCamber (
    float camberAtTop,
    float camberAtBottom )
```

Sets linear camber between the two values.

Parameters

<i>camberAtTop</i>	
<i>camberAtBottom</i>	

6.74.3.11 SetCamber() [2/3]

```
void NWH.WheelController3D.WheelController.SetCamber (
    float camber )
```

Sets fixed camber.

Parameters

<i>camber</i>	
---------------	--

6.74.3.12 SetCamber() [3/3]

```
void NWH.WheelController3D.WheelController.SetCamber (
    AnimationCurve curve )
```

Sets camber using AnimationCurve.

Parameters

<i>curve</i>	
--------------	--

6.74.3.13 Start()

```
void NWH.WheelController3D.WheelController.Start ( )
```

6.74.4 Member Data Documentation

6.74.4.1 activeFrictionPreset

[FrictionPreset](#) NWH.WheelController3D.WheelController.activeFrictionPreset

Current active friction preset.

6.74.4.2 activeFrictionPresetEnum

`FrictionPreset.FrictionPresetEnum` NWH.WheelController3D.WheelController.activeFrictionPreset←
Enum

Current active preset enum value.

6.74.4.3 applyForceToOthers

`bool` NWH.WheelController3D.WheelController.applyForceToOthers = false

6.74.4.4 damper

`Damper` NWH.WheelController3D.WheelController.damper

6.74.4.5 debug

`bool` NWH.WheelController3D.WheelController.debug

6.74.4.6 fFriction

`Friction` NWH.WheelController3D.WheelController.fFriction

Forward (longitudinal) friction info.

6.74.4.7 maxPutDownForce

`float` NWH.WheelController3D.WheelController.maxPutDownForce

6.74.4.8 parent

`GameObject` NWH.WheelController3D.WheelController.parent

Root object of the vehicle.

6.74.4.9 sFriction

`Friction` NWH.WheelController3D.WheelController.sFriction

Side (lateral) friction info.

6.74.4.10 singleRay

`bool` NWH.WheelController3D.WheelController.singleRay = false

6.74.4.11 singleWheelHit

`WheelHit` NWH.WheelController3D.WheelController.singleWheelHit = new `WheelHit`()

6.74.4.12 spring

`Spring` NWH.WheelController3D.WheelController.spring

6.74.4.13 trackedVehicle

`bool` NWH.WheelController3D.WheelController.trackedVehicle = false

Enables some wheel behaviors specific to tracked vehicles, specifically the fact that there is no wheel spin.

6.74.4.14 useRimCollider

`bool` NWH.WheelController3D.WheelController.useRimCollider

If enabled mesh collider mimicking the shape of rim and wheel will be positioned so that wheel can not pass through objects in case raycast does not detect the surface in time.

6.74.4.15 wheel

`Wheel` NWH.WheelController3D.WheelController.wheel

6.74.4.16 wheelHit

```
WheelHit NWH.WheelController3D.WheelController.wheelHit = new WheelHit()
```

Contains point in which wheel touches ground. Not valid if !isGrounded.

6.74.5 Property Documentation

6.74.5.1 angularVelocity

```
float NWH.WheelController3D.WheelController.angularVelocity [get]
```

Returns angular velocity of the wheel in radians. Multiply by wheel radius to get linear speed.

6.74.5.2 brakeTorque

```
float NWH.WheelController3D.WheelController.brakeTorque [get], [set]
```

Brake torque on the wheel axle. [Nm] Must be positive (zero included).

6.74.5.3 camber

```
float NWH.WheelController3D.WheelController.camber [get]
```

Camber angle of the wheel. [deg] Negative angle means that the top of the wheel is closer to the vehicle than the bottom.

6.74.5.4 damperCurve

```
AnimationCurve NWH.WheelController3D.WheelController.damperCurve [get], [set]
```

Damper force curve in relation to spring velocity.

6.74.5.5 damperForce

```
float NWH.WheelController3D.WheelController.damperForce [get]
```

Current damper force.

6.74.5.6 damperUnitBumpForce

`float NWH.WheelController3D.WheelController.damperUnitBumpForce [get], [set]`

Bump force at 1 m/s spring velocity

6.74.5.7 damperUnitReboundForce

`float NWH.WheelController3D.WheelController.damperUnitReboundForce [get], [set]`

Rebounding force at 1 m/s spring velocity

6.74.5.8 forwardFriction

`Friction NWH.WheelController3D.WheelController.forwardFriction [get], [set]`

Returns `_Friction` object with longitudinal values.

6.74.5.9 ForwardScanResolution

`int NWH.WheelController3D.WheelController.ForwardScanResolution [get], [set]`

Ground scan resolution in forward direction.

6.74.5.10 isGrounded

`bool NWH.WheelController3D.WheelController.isGrounded [get]`

Is the tractive surface touching the ground? Returns false if vehicle tipped over / tire sidewall is in contact.

6.74.5.11 mass

`float NWH.WheelController3D.WheelController.mass [get], [set]`

Mass of the wheel. [kg] Typical values would be in range [20, 200]

6.74.5.12 MaxPutDownForce

`float NWH.WheelController3D.WheelController.MaxPutDownForce [get]`

6.74.5.13 motorTorque

`float NWH.WheelController3D.WheelController.motorTorque [get], [set]`

Motor torque on the wheel axle. [Nm] Can be positive or negative based on direction.

6.74.5.14 NonRotating

`GameObject NWH.WheelController3D.WheelController.NonRotating [get], [set]`

Object that follows the wheel position in everything but rotation around the axle. Can be used for brake calipers, external fenders, etc.

6.74.5.15 Parent

`GameObject NWH.WheelController3D.WheelController.Parent [get], [set]`

Returns wheel's parent object.

6.74.5.16 pointVelocity

`Vector3 NWH.WheelController3D.WheelController.pointVelocity [get]`

Returns velocity at the wheel's center position in [m/s].

6.74.5.17 radius

`float NWH.WheelController3D.WheelController.radius [get], [set]`

Equal to tireRadis but exists because of compatibility with inbuilt WheelCollider. Radius of the complete wheel. [meters] Must be larger than 0.

6.74.5.18 rimOffset

`float NWH.WheelController3D.WheelController.rimOffset [get], [set]`

Side offset of the rim. Positive value will result if wheel further from the vehicle. [meters]

6.74.5.19 rpm

`float NWH.WheelController3D.WheelController.rpm [get]`

Rotations per minute of the wheel around the axle. [rpm]

6.74.5.20 ScanIgnoreLayers

`LayerMask NWH.WheelController3D.WheelController.ScanIgnoreLayers [get], [set]`

Layers that will be ignored when doing ground detection.

6.74.5.21 sideFriction

`Friction NWH.WheelController3D.WheelController.sideFriction [get], [set]`

Returns `_Friction` object with lateral values.

6.74.5.22 SideToSideScanResolution

`int NWH.WheelController3D.WheelController.SideToSideScanResolution [get], [set]`

Number of scan planes parallel to the wheel.

6.74.5.23 speed

`float NWH.WheelController3D.WheelController.speed [get]`

Returns vehicle speed in meters per second [m/s], multiply by 3.6 for [kph] or by 2.24 for [mph].

6.74.5.24 springBottomedOut

```
bool NWH.WheelController3D.WheelController.springBottomedOut [get]
```

True when spring is fully compressed, i.e. there is no more spring travel.

6.74.5.25 springCompression

```
float NWH.WheelController3D.WheelController.springCompression [get]
```

Returns value in range [0,1] where 1 means spring is fully compressed.

6.74.5.26 springCurve

```
AnimationCurve NWH.WheelController3D.WheelController.springCurve [get], [set]
```

[Spring](#) force curve in relation to spring length.

6.74.5.27 springLength

```
float NWH.WheelController3D.WheelController.springLength [get], [set]
```

Length of the spring when fully extended.

6.74.5.28 springMaximumForce

```
float NWH.WheelController3D.WheelController.springMaximumForce [get], [set]
```

Maximum spring force. [N]

6.74.5.29 springOverExtended

```
bool NWH.WheelController3D.WheelController.springOverExtended [get]
```

True when spring is fully extended.

6.74.5.30 springTravel

`float NWH.WheelController3D.WheelController.springTravel [get]`

Current length (travel) of spring.

6.74.5.31 springTravelPoint

`Vector3 NWH.WheelController3D.WheelController.springTravelPoint [get]`

Point in which spring and swingarm are in contact.

6.74.5.32 springVelocity

`float NWH.WheelController3D.WheelController.springVelocity [get]`

[Spring](#) velocity in relation to local vertical axis. [m/s] Positive on rebound (extension), negative on bump (compression).

6.74.5.33 steerAngle

`float NWH.WheelController3D.WheelController.steerAngle [get], [set]`

Steer angle around the wheel's up axis (with add-ons ignored). [deg]

6.74.5.34 suspensionForce

`float NWH.WheelController3D.WheelController.suspensionForce [get], [set]`

Current spring force. [N] Can be written to for use in Anti-roll Bar script or similar.

6.74.5.35 tireRadius

`float NWH.WheelController3D.WheelController.tireRadius [get], [set]`

Radius (height) of the tire. [meters]

6.74.5.36 tireWidth

`float NWH.WheelController3D.WheelController.tireWidth [get], [set]`

Width of the wheel. [meters]

6.74.5.37 VehicleSide

`Side NWH.WheelController3D.WheelController.VehicleSide [get], [set]`

Returns Enum [Side] with the corresponding side of the vehicle a wheel is at [Left, Right]

6.74.5.38 Visual

`GameObject NWH.WheelController3D.WheelController.Visual [get], [set]`

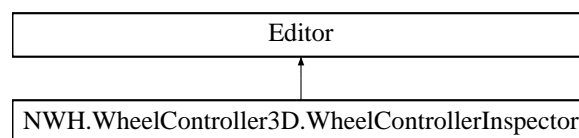
Returns object that represents wheel's visual representation. Can be null in case the object is not assigned (not mandatory).

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/[WheelController.API.cs](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/[WheelController.cs](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/[WheelController.Debug.cs](#)

6.75 NWH.WheelController3D.WheelControllerInspector Class Reference

Inheritance diagram for NWH.WheelController3D.WheelControllerInspector:



Public Member Functions

- override void [OnInspectorGUI](#) ()

Public Attributes

- AnimationCurve [forwardFrictionCurve](#)
- AnimationCurve [sideFrictionCurve](#)
- Vector4 [sideParams](#) = new Vector4()
- Vector4 [forwardParams](#) = new Vector4()

6.75.1 Member Function Documentation

6.75.1.1 OnInspectorGUI()

```
override void NWH.WheelController3D.WheelControllerInspector.OnInspectorGUI ( )
```

6.75.2 Member Data Documentation

6.75.2.1 forwardFrictionCurve

```
AnimationCurve NWH.WheelController3D.WheelControllerInspector.forwardFrictionCurve
```

6.75.2.2 forwardParams

```
Vector4 NWH.WheelController3D.WheelControllerInspector.forwardParams = new Vector4()
```

6.75.2.3 sideFrictionCurve

```
AnimationCurve NWH.WheelController3D.WheelControllerInspector.sideFrictionCurve
```

6.75.2.4 sideParams

```
Vector4 NWH.WheelController3D.WheelControllerInspector.sideParams = new Vector4()
```

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/Editor/[WheelControllerInspector.cs](#)

6.76 NWH.WheelController3D.WheelController.WheelHit Class Reference

Contains RaycastHit and extended hit data.

Public Member Functions

- [WheelHit](#) ()
- void [Copy](#) ([WheelHit](#) hit, bool copyHit)

Public Attributes

- RaycastHit [raycastHit](#)
- float [angleForward](#)
- float [distanceFromTire](#)
- Vector2 [offset](#)
- float [weight](#)
- bool [valid](#) = false
- float [curvatureOffset](#)
- Vector3 [groundPoint](#)
- Vector3 [forwardDir](#)
The direction the wheel is pointing in.
- float [forwardSlip](#)
Tire slip in the rolling direction.
- Vector3 [sidewaysDir](#)
The sideways direction of the wheel.
- float [sidewaysSlip](#)
The slip in the sideways direction.
- float [force](#)
The magnitude of the force being applied for the contact. [N]

Properties

- Vector3 [point](#) [get]
The point of contact between the wheel and the ground.
- Vector3 [normal](#) [get]
The normal at the point of contact
- Collider [collider](#) [get]

6.76.1 Detailed Description

Contains RaycastHit and extended hit data.

6.76.2 Constructor & Destructor Documentation

6.76.2.1 WheelHit()

```
NWH.WheelController3D.WheelController.WheelHit.WheelHit ( )
```


6.76.3 Member Function Documentation

6.76.3.1 Copy()

```
void NWH.WheelController3D.WheelController.WheelHit.Copy (
    WheelHit hit,
    bool copyHit )
```

6.76.4 Member Data Documentation

6.76.4.1 angleForward

```
float NWH.WheelController3D.WheelController.WheelHit.angleForward
```

6.76.4.2 curvatureOffset

```
float NWH.WheelController3D.WheelController.WheelHit.curvatureOffset
```

6.76.4.3 distanceFromTire

```
float NWH.WheelController3D.WheelController.WheelHit.distanceFromTire
```

6.76.4.4 force

```
float NWH.WheelController3D.WheelController.WheelHit.force
```

The magnitude of the force being applied for the contact. [N]

6.76.4.5 forwardDir

```
Vector3 NWH.WheelController3D.WheelController.WheelHit.forwardDir
```

The direction the wheel is pointing in.

6.76.4.6 forwardSlip

```
float NWH.WheelController3D.WheelController.WheelHit.forwardSlip
```

Tire slip in the rolling direction.

6.76.4.7 groundPoint

```
Vector3 NWH.WheelController3D.WheelController.WheelHit.groundPoint
```

6.76.4.8 offset

```
Vector2 NWH.WheelController3D.WheelController.WheelHit.offset
```

6.76.4.9 raycastHit

```
RaycastHit NWH.WheelController3D.WheelController.WheelHit.raycastHit
```

6.76.4.10 sidewaysDir

```
Vector3 NWH.WheelController3D.WheelController.WheelHit.sidewaysDir
```

The sideways direction of the wheel.

6.76.4.11 sidewaysSlip

```
float NWH.WheelController3D.WheelController.WheelHit.sidewaysSlip
```

The slip in the sideways direction.

6.76.4.12 valid

```
bool NWH.WheelController3D.WheelController.WheelHit.valid = false
```

6.76.4.13 weight

`float NWH.WheelController3D.WheelController.WheelHit.weight`

6.76.5 Property Documentation

6.76.5.1 collider

`Collider NWH.WheelController3D.WheelController.WheelHit.collider [get]`

6.76.5.2 normal

`Vector3 NWH.WheelController3D.WheelController.WheelHit.normal [get]`

The normal at the point of contact

6.76.5.3 point

`Vector3 NWH.WheelController3D.WheelController.WheelHit.point [get]`

The point of contact between the wheel and the ground.

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

- E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/[WheelController.cs](#)

Chapter 7

File Documentation

7.1 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraChanger.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.CameraChanger](#)

Switches between the camera objects that are children to this object and contain camera tag, in order they appear in the hierarchy or in order they are added to the vehicle cameras list.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.2 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraFollow.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.CameraFollow](#)

Camera that follows behind the vehicle.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.3 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraInsideVehicle.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.CameraInsideVehicle](#)

Empty component that should be attached to the cameras that are inside the vehicle if interior sound change is to be used.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.4 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraMouseDrag.cs File Reference

Classes

- class [NWH.VehiclePhysics.CameraMouseDrag](#)

Camera that can be dragged with the mouse.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.5 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Cameras/CameraOnboard.cs File Reference

Classes

- class [NWH.VehiclePhysics.CameraOnboard](#)

Camera for on or in-vehicle use with option of head movement according to the G-forces.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.6 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Axle.cs File Reference

Classes

- class [NWH.VehiclePhysics.Axle](#)
Represents a single axle of a vehicle.
- class [NWH.VehiclePhysics.Axle.Geometry](#)

Class holding all geometry related data for axle and it's wheels.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.7 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Brakes.cs File Reference

Classes

- class [NWH.VehiclePhysics.Brakes](#)

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.8 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DamageHandler.cs File Reference

Classes

- class [NWH.VehiclePhysics.DamageHandler](#)
Handles all damage related calculations and mesh deformations. Collision sounds are handled by [CrashComponent](#) class.
- class [NWH.VehiclePhysics.DamageHandler.CollisionEvent](#)
Contains data on the collision that has last happened.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.9 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/ABS.cs File Reference

Classes

- class [NWH.VehiclePhysics.ABS](#)
Traction control class.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.10 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/DrivingAssists.cs File Reference

Classes

- class [NWH.VehiclePhysics.DrivingAssists](#)
Systems to help driver with vehicle control.
- class [NWH.VehiclePhysics.DrivingAssists.DrivingAid](#)
Base class for driving aids.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.11 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/Stability.cs File Reference ↩↪

Classes

- class [NWH.VehiclePhysics.Stability](#)
Traction control class.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.12 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/DrivingAssists/TractionControl.cs File Reference ↩↪

Classes

- class [NWH.VehiclePhysics.TractionControl](#)
Traction control class.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.13 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Editor/VehicleControllerInspector.cs File Reference ↩↪

Classes

- class [NWH.VehiclePhysics.VehicleControllerInspector](#)
Shows different settings for trailer depending if vehicle is a trailer or a towing vehicle.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.14 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/Backfire.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.Backfire](#)
Controls exhaust flames / flashes.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.15 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/Effects.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.Effects](#)
Main class for handling visual effects such as skidmarks, lights and exhausts.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.16 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/ExhaustSmoke.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.ExhaustSmoke](#)
Controls particle emitters that represent exhausts.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.17 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/Lights.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.Lights](#)
Class for controlling all of the vehicle lights.
- class [NWH.VehiclePhysics.Lights.VehicleLight](#)
Single vehicle light.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.18 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/↔ SkidmarkDestroy.cs File Reference

Classes

- class [NWH.VehiclePhysics.SkidmarkDestroy](#)
Destroys skidmark object when distance to the vehicle becomes greater then distance threshold.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.19 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/↔ SkidmarkGenerator.cs File Reference

Classes

- class [NWH.VehiclePhysics.SkidmarkGenerator](#)
Generates skidmark meshes.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.20 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/↔ Skidmarks.cs File Reference

Classes

- class [NWH.VehiclePhysics.Skidmarks](#)

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.21 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Effects/SurfaceParticles.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.SurfaceParticles](#)
Skid smoke and surface dust generated by wheel slipping / rolling over the surface.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.22 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Engine.cs File Reference

Classes

- class [NWH.VehiclePhysics.Engine](#)
Everything related to a vehicle's engine and it's systems.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.23 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/FlipOver.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.FlipOver](#)

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.24 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/ForcedInduction.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.ForcedInduction](#)
Supercharger, turbocharger, etc. Can also be used on vehicles with no forced induction for sound effects such as intake noise or engine fan noise.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.25 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Fuel.cs File Reference

Classes

- class [NWH.VehiclePhysics.Fuel](#)

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.26 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Ground↔Detection.cs File Reference

Classes

- class [NWH.VehiclePhysics.GroundDetection](#)
- class [NWH.VehiclePhysics.GroundDetection.GroundEntity](#)
A class representing a single ground surface type.
- class [NWH.VehiclePhysics.GroundDetection.SoundComponent](#)
Base class for surface sounds.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.27 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/↔DesktopInputManager.cs File Reference

Classes

- class [NWH.VehiclePhysics.DesktopInputManager](#)
Class for handling desktop user input via mouse and keyboard. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.28 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/↔ InputStates.cs File Reference

Classes

- class [NWH.VehiclePhysics.InputStates](#)
Class for storing input states of the vehicle.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.29 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/↔ MobileInputManager.cs File Reference

Classes

- class [NWH.VehiclePhysics.MobileInputManager](#)
Class for handling mobile user input via touch screen and sensors. Avoid having two input managers active at the same time (mobile and desktop) as the last executed script will override the first one.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.30 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Input/↔ SteeringWheel.cs File Reference

Classes

- class [NWH.VehiclePhysics.SteeringWheel](#)
Script for controlling the GUI steering wheel for mobile controls. Credits go to yasirkula from Unity Forums for original code.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.31 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Metrics.cs File Reference

Classes

- class [NWH.VehiclePhysics.Metrics](#)
Class for holding metrics such as odometer, top speed and drift time. All the speed values are in m/s. If you need the value in km/h or mph use UnitConverter functions.
- class [NWH.VehiclePhysics.Metrics.Metric](#)

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.32 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/↔ CenterOfMass.cs File Reference

Classes

- class [NWH.VehiclePhysics.CenterOfMass](#)
Used for adjusting center of mass of any rigidbody object.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.33 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/↔ CharacterVehicleChanger.cs File Reference

Classes

- class [NWH.VehiclePhysics.CharacterVehicleChanger](#)
Allows character to enter or exit vehicle. Can be used with any first or 3rd person object.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.34 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/↔ Downforce.cs File Reference

Classes

- class [NWH.VehiclePhysics.Downforce](#)
Loose approximation on downforce acting on a vehicle. [Downforce](#) in this case is only dependent on speed, but its amount and speed at which it is achieved can be adjusted. Can be adjusted at runtime.
- class [NWH.VehiclePhysics.Downforce.DownforcePoint](#)
Single point at which downforce will be applied.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.35 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/RenderCameraDisable.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.RenderCameraDisable](#)
Attach this to any cameras rendering to mirrors so that when vehicle is inactive mirror will not update.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.36 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/UnitConverter.cs File Reference ↩

Classes

- class **NWH.VehiclePhysics.UnitConverter**

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.37 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/VehicleChanger.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.VehicleChanger](#)
Used for chaning vehicles. Also activates and deactivates vehicle cameras if default VehicleCamera system is used.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.38 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Misc/VehicleFromScript.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.VehicleFromScript](#)
Example script for setting up a vehicle at runtime - modify as/if needed. Sets up a vehicle from script at runtime. Model is required to already have body colliders set up, as well as wheels tagged with correct tag. Model also needs to have correct rotation (Z-forward, Y-up, X-right). Works only on vehicles with two wheels per axle. Number of axles is not limited.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.39 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Rigging.cs File Reference

Classes

- class [NWH.VehiclePhysics.Rigging](#)

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.40 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↔ AirBrakeComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.AirBrakeComponent](#)

Hiss produced by air brakes releasing air. Accepts multiple clips of which one will be chosen at random each time this effect is played.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.41 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↔ BackfireComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.BackfireComponent](#)

Exhaust popping on deceleration / rev limiter.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.42 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↔ BlinkerComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.BlinkerComponent](#)

Click-clack of the working blinker. Accepts two clips, first is for the blinker turning on and the second is for blinker turning off.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.43 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↔ CrashComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.CrashComponent](#)

[Sound](#) of vehicle crashing into an object. Supports multiple audio clips of which one will be chosen at random each time this effect is played.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.44 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↔ EngineIdleComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.EngineIdleSoundComponent](#)

[Sound](#) of an engine idling.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.45 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↔ EngineStartStopComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.EngineStartStopComponent](#)

[Sound](#) of an engine starting / stopping. First audio clip is for engine starting, and second one is for engine stopping.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.46 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↵ GearChangeComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.GearChangeComponent](#)
Shifter sound played when changing gears. Supports multiple audio clips of which one is chosen at random each time this effect is played.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.47 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↵ HornComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.HornComponent](#)
Vehicle horn sound.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.48 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↵ SkidComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.SkidComponent](#)
[Sound](#) produced by tire skidding over surface.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.49 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/Sound.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.Sound](#)
Main class that manages all the sound aspects of the vehicle.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.50 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/SoundComponent.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.SoundComponent](#)
Base abstract class from which all vehicle sound components inherit.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.51 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/SurfaceComponent.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.SurfaceComponent](#)
Sounds produced by tire rolling over the surface.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.52 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/SuspensionComponent.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.SuspensionComponent](#)
Sound of wheel hitting the surface or obstracle.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.53 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↔ TransmissionWhineComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.TransmissionWhineComponent](#)

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.54 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↔ TurboFlutterComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.TurboFlutterComponent](#)
Sound of wastegate releasing air on turbocharged vehicles.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.55 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Sound/↔ TurboWhistleComponent.cs File Reference

Classes

- class [NWH.VehiclePhysics.TurboWhistleComponent](#)
Sound of turbocharger or supercharger.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.56 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Steering.cs File Reference

Classes

- class [NWH.VehiclePhysics.Steering](#)
Everything related to steering and axle's geometry.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.57 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Tracks.cs File Reference

Classes

- class [NWH.VehiclePhysics.Tracks](#)
Class for handling tracked vehicles. If enabled all the wheels should be the same size and placed on either left or right side of the vehicle. Center wheels are not supported in this mode.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.58 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Trailer↔ Handler.cs File Reference

Classes

- class [NWH.VehiclePhysics.TrailerHandler](#)
Everthing related to a trailer.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.59 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Transmission.cs File Reference

Classes

- class [NWH.VehiclePhysics.Transmission](#)
Handles gear changing and also torque and RPM transmission in both directions.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.60 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/VehicleController.cs File Reference ↩

Classes

- class [NWH.VehiclePhysics.VehicleController](#)
Main class controlling all the other parts of the vehicle.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.61 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Vehicle/Wheel.cs File Reference

Classes

- class [NWH.VehiclePhysics.Wheel](#)
Contains everything related to wheels. To access WC3D's properties directly use WheelController getter. Most used variables are wrapped in getters and setter to enable eventual (but unlikely) future compatibility with default wheel collider. [Wheel](#) class is not equal to WheelController class. To access WC3D (WheelController) use WheelController getter/setter.

Namespaces

- namespace [NWH.VehiclePhysics](#)

7.62 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/Editor/WheelControllerInspector.cs File Reference ↩

Classes

- class [NWH.WheelController3D.WheelControllerInspector](#)

Namespaces

- namespace [NWH.WheelController3D](#)

7.63 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/WheelController.API.cs File Reference ↩

Classes

- class [NWH.WheelController3D.WheelController](#)
API for [WheelController](#)

Namespaces

- namespace [NWH.WheelController3D](#)

7.64 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/WheelController.cs File Reference ↩

Classes

- class [NWH.WheelController3D.WheelController](#)
API for [WheelController](#)
- class [NWH.WheelController3D.WheelController.Friction](#)
All info related to longitudinal force calculation.
- class [NWH.WheelController3D.WheelController.Damper](#)
Suspension part.
- class [NWH.WheelController3D.WheelController.Spring](#)
Suspension part.
- class [NWH.WheelController3D.WheelController.Wheel](#)
Contains everything wheel related, including rim and tire.
- class [NWH.WheelController3D.WheelController.WheelHit](#)
Contains RaycastHit and extended hit data.

Namespaces

- namespace [NWH.WheelController3D](#)

7.65 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/WheelController.Debug.cs File Reference ↩

Classes

- class [NWH.WheelController3D.WheelController](#)
API for [WheelController](#)

Namespaces

- namespace [NWH.WheelController3D](#)

7.66 E:/UnitySoft/NWH Vehicle Physics/Assets/VehiclePhysics/Scripts/Wheel/WheelController.FrictionPreset.cs File Reference ↩

Classes

- class [NWH.WheelController3D.WheelController](#)
API for [WheelController](#)
- class [NWH.WheelController3D.WheelController.FrictionPreset](#)
Container class for holding wheel friction presets.

Namespaces

- namespace [NWH.WheelController3D](#)

Index

- abs
 - NWH::VehiclePhysics::DrivingAssists, [52](#)
- Acceleration
 - NWH::VehiclePhysics::VehicleController, [168](#)
- acceleration
 - NWH::WheelController3D::WheelController::↔
Wheel, [185](#)
- ackermannPercent
 - NWH::VehiclePhysics::Axle::Geometry, [80](#)
- Activate
 - NWH::VehiclePhysics::Wheel, [177](#)
- Active
 - NWH::VehiclePhysics::Brakes, [27](#)
 - NWH::VehiclePhysics::VehicleController, [168](#)
- active
 - NWH::VehiclePhysics::DrivingAssists::DrivingAid, [50](#)
 - NWH::VehiclePhysics::Lights::VehicleLight, [174](#)
- activeFrictionPreset
 - NWH::WheelController3D::WheelController, [195](#)
- activeFrictionPresetEnum
 - NWH::WheelController3D::WheelController, [195](#)
- ActiveVehicleController
 - NWH::VehiclePhysics::VehicleChanger, [159](#)
- AddBrakeTorque
 - NWH::VehiclePhysics::Wheel, [177](#)
- AddedClutchRPM
 - NWH::VehiclePhysics::Transmission, [150](#)
- AdjustGeometry
 - NWH::VehiclePhysics::Steering, [127](#)
- AdjustedShiftDownRpm
 - NWH::VehiclePhysics::Transmission, [150](#)
- AdjustedShiftUpRpm
 - NWH::VehiclePhysics::Transmission, [150](#)
- airBrakeComponent
 - NWH::VehiclePhysics::Sound, [113](#)
- airBrakePressure
 - NWH::VehiclePhysics::Brakes, [26](#)
- airBrakes
 - NWH::VehiclePhysics::Brakes, [26](#)
- allowedAngle
 - NWH::VehiclePhysics::FlipOver, [67](#)
- allowedDamage
 - NWH::VehiclePhysics::DamageHandler, [42](#)
- amount
 - NWH::VehiclePhysics::Fuel, [77](#)
- Angle
 - NWH::VehiclePhysics::Steering, [129](#)
- angleFollowStrength
 - NWH::VehiclePhysics::CameraFollow, [30](#)
- angleForward
 - NWH::WheelController3D::WheelController::↔
WheelHit, [207](#)
- AngleSigned
 - NWH::VehiclePhysics::CameraFollow, [29](#)
 - NWH::VehiclePhysics::VehicleController, [162](#)
- angularVelocity
 - NWH::WheelController3D::WheelController, [198](#)
 - NWH::WheelController3D::WheelController::↔
Wheel, [185](#)
- antiRollBarForce
 - NWH::VehiclePhysics::Axle::Geometry, [81](#)
- applyForceToOthers
 - NWH::WheelController3D::WheelController, [196](#)
- Arcade
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [74](#)
- attachDistanceThreshold
 - NWH::VehiclePhysics::TrailerHandler, [140](#)
- attachOnPlay
 - NWH::VehiclePhysics::TrailerHandler, [140](#)
- attached
 - NWH::VehiclePhysics::TrailerHandler, [140](#)
- attachmentPoint
 - NWH::VehiclePhysics::TrailerHandler, [140](#)
- audioMixerGroup
 - NWH::VehiclePhysics::SoundComponent, [121](#)
- automaticClutch
 - NWH::VehiclePhysics::Transmission, [147](#)
- averageSpeed
 - NWH::VehiclePhysics::Metrics, [96](#)
- axleBones
 - NWH::VehiclePhysics::Rigging, [102](#)
- axles
 - NWH::VehiclePhysics::VehicleController, [165](#)
- BCDE
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [74](#)
- baseIntensity
 - NWH::VehiclePhysics::ExhaustSmoke, [65](#)
- Bias
 - NWH::VehiclePhysics::Axle, [20](#)
 - NWH::VehiclePhysics::Wheel, [179](#)
- blinkerComponent
 - NWH::VehiclePhysics::Sound, [113](#)
- BlinkerState
 - NWH::VehiclePhysics::Lights, [94](#)
- bottomOutForce

- NWH::WheelController3D::WheelController::↔
Spring, 124
- bottomedOut
 - NWH::WheelController3D::WheelController::↔
Spring, 123
- brakeCoefficient
 - NWH::VehiclePhysics::Axle, 19
 - NWH::VehiclePhysics::Wheel, 179
- BrakeDown
 - NWH::VehiclePhysics::MobileInputManager, 98
- BrakeTorque
 - NWH::VehiclePhysics::Wheel, 179
- brakeTorque
 - NWH::WheelController3D::WheelController, 198
 - NWH::WheelController3D::WheelController::↔
Wheel, 185
- BrakeUp
 - NWH::VehiclePhysics::MobileInputManager, 98
- brakeWhileIdle
 - NWH::VehiclePhysics::Brakes, 26
- brakes
 - NWH::VehiclePhysics::VehicleController, 165
- breakForce
 - NWH::VehiclePhysics::TrailerHandler, 140
- camber
 - NWH::WheelController3D::WheelController, 198
- camberAngle
 - NWH::WheelController3D::WheelController::↔
Wheel, 185
- camberAtBottom
 - NWH::VehiclePhysics::Axle::Geometry, 81
- camberAtTop
 - NWH::VehiclePhysics::Axle::Geometry, 81
- camberCurve
 - NWH::WheelController3D::WheelController::↔
Wheel, 185
- cameraTag
 - NWH::VehiclePhysics::CameraChanger, 28
- CanShift
 - NWH::VehiclePhysics::Transmission, 151
- capacity
 - NWH::VehiclePhysics::Fuel, 77
- casterAngle
 - NWH::VehiclePhysics::Axle::Geometry, 81
- centerOfMassOffset
 - NWH::VehiclePhysics::CenterOfMass, 36
- ChangeCamera
 - NWH::VehiclePhysics::MobileInputManager, 99
- ChangeVehicle
 - NWH::VehiclePhysics::MobileInputManager, 99
 - NWH::VehiclePhysics::VehicleChanger, 157, 158
- CharacterBased
 - NWH::VehiclePhysics::VehicleChanger, 159
- characterControllerObject
 - NWH::VehiclePhysics::CharacterVehicleChanger, 37
- ClampAngle
 - NWH::VehiclePhysics::CameraMouseDrag, 32
- Clip
 - NWH::VehiclePhysics::SoundComponent, 122
- clip
 - NWH::VehiclePhysics::GroundDetection::Sound↔
Component, 117
- Clips
 - NWH::VehiclePhysics::SoundComponent, 122
- clips
 - NWH::VehiclePhysics::SoundComponent, 121
- Clutch
 - NWH::VehiclePhysics::InputStates, 90
- clutchEngagementCurve
 - NWH::VehiclePhysics::Transmission, 147
- clutchPedalPressedPercent
 - NWH::VehiclePhysics::Transmission, 148
- ClutchPercent
 - NWH::VehiclePhysics::Transmission, 151
- collider
 - NWH::WheelController3D::WheelController::↔
WheelHit, 209
- collision
 - NWH::VehiclePhysics::DamageHandler::Collision↔
Event, 39
- CollisionInfo
 - NWH::VehiclePhysics::VehicleController, 168
- CollisionState
 - NWH::VehiclePhysics::VehicleController, 168
- compressionPercent
 - NWH::WheelController3D::WheelController::↔
Spring, 124
- ConsumptionKilometersPerLiter
 - NWH::VehiclePhysics::Fuel, 78
- ConsumptionLitersPer100Kilometers
 - NWH::VehiclePhysics::Fuel, 78
- ConsumptionLitersPerSecond
 - NWH::VehiclePhysics::Fuel, 78
- ConsumptionMPG
 - NWH::VehiclePhysics::Fuel, 78
- continousDriftDistance
 - NWH::VehiclePhysics::Metrics, 96
- continousDriftTime
 - NWH::VehiclePhysics::Metrics, 96
- ControllerGO
 - NWH::VehiclePhysics::Wheel, 179
- ControllerTransform
 - NWH::VehiclePhysics::Wheel, 179
- Copy
 - NWH::WheelController3D::WheelController::↔
WheelHit, 207
- crashComponent
 - NWH::VehiclePhysics::Sound, 113
- crashMixerGroup
 - NWH::VehiclePhysics::Sound, 113
- CreateNewSnapshot
 - NWH::VehiclePhysics::SkidmarkGenerator, 106
- crossoverSpeed
 - NWH::VehiclePhysics::Steering, 128
- currentCamera

- NWH::VehiclePhysics::CameraChanger, [28](#)
- CurrentGroundEntity
 - NWH::VehiclePhysics::Wheel, [180](#)
- CurrentGroundEntityName
 - NWH::VehiclePhysics::Wheel, [180](#)
- CurrentVehicle
 - NWH::VehiclePhysics::VehicleChanger, [159](#)
- curvatureOffset
 - NWH::WheelController3D::WheelController::↔
WheelHit, [207](#)
- Curve
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [76](#)
- Damage
 - NWH::VehiclePhysics::DamageHandler, [44](#)
 - NWH::VehiclePhysics::Wheel, [180](#)
- damage
 - NWH::VehiclePhysics::VehicleController, [165](#)
- DamagePercent
 - NWH::VehiclePhysics::DamageHandler, [44](#)
- DamageSteerDirection
 - NWH::VehiclePhysics::Wheel, [180](#)
- damper
 - NWH::WheelController3D::WheelController, [196](#)
- damperCurve
 - NWH::WheelController3D::WheelController, [198](#)
- damperForce
 - NWH::WheelController3D::WheelController, [198](#)
- damperUnitBumpForce
 - NWH::WheelController3D::WheelController, [198](#)
- damperUnitReboundForce
 - NWH::WheelController3D::WheelController, [199](#)
- dampingCurve
 - NWH::WheelController3D::WheelController::↔
Damper, [45](#)
- DeactivateAll
 - NWH::VehiclePhysics::VehicleChanger, [159](#)
- debug
 - NWH::WheelController3D::WheelController, [196](#)
- decelerationMagnitude
 - NWH::VehiclePhysics::DamageHandler::Collision↔
Event, [39](#)
- decelerationThreshold
 - NWH::VehiclePhysics::DamageHandler, [43](#)
- deformationQueue
 - NWH::VehiclePhysics::DamageHandler::Collision↔
Event, [39](#)
- deformationRadius
 - NWH::VehiclePhysics::DamageHandler, [43](#)
- deformationRandomness
 - NWH::VehiclePhysics::DamageHandler, [43](#)
- deformationStrength
 - NWH::VehiclePhysics::DamageHandler, [43](#)
- deformationVerticesPerFrame
 - NWH::VehiclePhysics::DamageHandler, [43](#)
- degreesPerSecondLimit
 - NWH::VehiclePhysics::Steering, [128](#)
- DetectWheelSkid
 - NWH::VehiclePhysics::VehicleController, [163](#)
- DetectWheelSpin
 - NWH::VehiclePhysics::VehicleController, [163](#)
- DetermineSide
 - NWH::WheelController3D::WheelController, [193](#)
- differentialStrength
 - NWH::VehiclePhysics::Axle, [19](#)
- DifferentialType
 - NWH::VehiclePhysics::Axle, [18](#)
 - NWH::VehiclePhysics::Transmission, [144](#)
- differentialType
 - NWH::VehiclePhysics::Axle, [19](#)
 - NWH::VehiclePhysics::Transmission, [148](#)
- Direction
 - NWH::VehiclePhysics::VehicleController, [168](#)
- Disable
 - NWH::VehiclePhysics::Sound, [111](#)
 - NWH::VehiclePhysics::SoundComponent, [120](#)
- distance
 - NWH::VehiclePhysics::CameraFollow, [30](#)
 - NWH::VehiclePhysics::CameraMouseDrag, [32](#)
- distanceFromTire
 - NWH::WheelController3D::WheelController::↔
WheelHit, [207](#)
- distanceSmoothing
 - NWH::VehiclePhysics::CameraMouseDrag, [32](#)
- distanceThreshold
 - NWH::VehiclePhysics::SkidmarkDestroy, [105](#)
- DoubleSubArray
 - NWH::VehiclePhysics::SkidmarkGenerator, [106](#)
- downforcePoints
 - NWH::VehiclePhysics::Downforce, [48](#)
- DragEvent
 - NWH::VehiclePhysics::SteeringWheel, [130](#)
- drivingAids
 - NWH::VehiclePhysics::VehicleController, [165](#)
- duration
 - NWH::VehiclePhysics::Backfire, [22](#)
- dust
 - NWH::VehiclePhysics::SurfaceParticles, [133](#)
- dustColor
 - NWH::VehiclePhysics::GroundDetection::Ground↔
Entity, [85](#)
- dustGo
 - NWH::VehiclePhysics::SurfaceParticles, [133](#)
- dustIntensity
 - NWH::VehiclePhysics::GroundDetection::Ground↔
Entity, [85](#)
- dustPrefab
 - NWH::VehiclePhysics::GroundDetection, [83](#)
- dynamicShiftPoint
 - NWH::VehiclePhysics::Transmission, [148](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Cameras/CameraChanger.cs, [211](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Cameras/CameraFollow.cs, [211](#)

- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Cameras/CameraInside↔
Vehicle.cs, [211](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Cameras/CameraMouse↔
Drag.cs, [212](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Cameras/CameraOnboard.↔
cs, [212](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Axle.cs, [212](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Brakes.cs, [213](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/DamageHandler.cs,
[213](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/DrivingAssists/AB↔
S.cs, [213](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/DrivingAssists/↔
DrivingAssists.cs, [213](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/DrivingAssists/↔
Stability.cs, [214](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/DrivingAssists/↔
TractionControl.cs, [214](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Editor/Vehicle↔
ControllerInspector.cs, [214](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Effects/Backfire.cs,
[215](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Effects/Effects.cs, [215](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Effects/Exhaust↔
Smoke.cs, [215](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Effects/Lights.cs, [215](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Effects/Skidmark↔
Destroy.cs, [216](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Effects/Skidmark↔
Generator.cs, [216](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Effects/Skidmarks.cs,
[216](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Effects/Surface↔
Particles.cs, [217](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Engine.cs, [217](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/FlipOver.cs, [217](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/ForcedInduction.cs,
[217](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Fuel.cs, [218](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/GroundDetection.cs,
[218](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Input/DesktopInput↔
Manager.cs, [218](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Input/InputStates.cs,
[219](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Input/MobileInput↔
Manager.cs, [219](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Input/Steering↔
Wheel.cs, [219](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Metrics.cs, [219](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Misc/CenterOf↔
Mass.cs, [220](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Misc/Character↔
VehicleChanger.cs, [220](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Misc/Downforce.cs,
[220](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Misc/Render↔
CameraDisable.cs, [221](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Misc/UnitConverter.↔
cs, [221](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Misc/Vehicle↔
Changer.cs, [221](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Misc/VehicleFrom↔
Script.cs, [221](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Rigging.cs, [222](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Sound/AirBrake↔
Component.cs, [222](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Sound/Backfire↔
Component.cs, [222](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Sound/Blinker↔
Component.cs, [223](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Sound/Crash↔
Component.cs, [223](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
Physics/Scripts/Vehicle/Sound/EngineIdle↔

- Component.cs, [223](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/EngineStart↔
 - StopComponent.cs, [223](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/GearChange↔
 - Component.cs, [224](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/HornComponent.↔
 - cs, [224](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/SkidComponent.↔
 - cs, [224](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/Sound.cs, [225](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/Sound↔
 - Component.cs, [225](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/Surface↔
 - Component.cs, [225](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/Suspension↔
 - Component.cs, [225](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/Transmission↔
 - WhineComponent.cs, [226](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/TurboFlutter↔
 - Component.cs, [226](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Sound/Turbo↔
 - WhistleComponent.cs, [226](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Steering.cs, [227](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Tracks.cs, [227](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/TrailerHandler.cs, [227](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Transmission.cs, [227](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/VehicleController.cs, [228](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Vehicle/Wheel.cs, [228](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Wheel/Editor/WheelController↔
 - Inspector.cs, [228](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Wheel/WheelController.AP↔
 - l.cs, [229](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Wheel/WheelController.cs, [229](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Wheel/WheelController.↔
 - Debug.cs, [229](#)
- E:/UnitySoft/NWH Vehicle Physics/Assets/Vehicle↔
 - Physics/Scripts/Wheel/WheelController.↔
 - FrictionPreset.cs, [230](#)
- effects
 - NWH::VehiclePhysics::VehicleController, [166](#)
- efficiency
 - NWH::VehiclePhysics::Fuel, [77](#)
- emit
 - NWH::VehiclePhysics::ExhaustSmoke, [65](#)
- Enable
 - NWH::VehiclePhysics::Sound, [111](#)
 - NWH::VehiclePhysics::SoundComponent, [120](#)
- enabled
 - NWH::VehiclePhysics::DamageHandler, [43](#)
 - NWH::VehiclePhysics::DrivingAssists::DrivingAid, [50](#)
 - NWH::VehiclePhysics::FlipOver, [67](#)
 - NWH::VehiclePhysics::Lights, [92](#)
 - NWH::VehiclePhysics::Rigging, [102](#)
 - NWH::VehiclePhysics::Skidmarks, [108](#)
- engine
 - NWH::VehiclePhysics::VehicleController, [166](#)
- engineIdleComponent
 - NWH::VehiclePhysics::Sound, [113](#)
- engineMixerGroup
 - NWH::VehiclePhysics::Sound, [113](#)
- EngineStartStop
 - NWH::VehiclePhysics::MobileInputManager, [99](#)
- engineStartStopComponent
 - NWH::VehiclePhysics::Sound, [113](#)
- Enqueue
 - NWH::VehiclePhysics::DamageHandler, [42](#)
- enterDistance
 - NWH::VehiclePhysics::CharacterVehicleChanger, [37](#)
- enterExitTag
 - NWH::VehiclePhysics::CharacterVehicleChanger, [37](#)
- exhaustFlash
 - NWH::VehiclePhysics::Effects, [53](#)
- exhaustPopComponent
 - NWH::VehiclePhysics::Sound, [114](#)
- exhausts
 - NWH::VehiclePhysics::Effects, [53](#)
- fFriction
 - NWH::WheelController3D::WheelController, [196](#)
- FixedUpdate
 - NWH::WheelController3D::WheelController, [193](#)
- Flash
 - NWH::VehiclePhysics::Backfire, [22](#)
- flipOver
 - NWH::VehiclePhysics::VehicleController, [166](#)
- flippedOver
 - NWH::VehiclePhysics::FlipOver, [67](#)
- flutterSoundFlag
 - NWH::VehiclePhysics::ForcedInduction, [69](#)
- followTargetsRotation
 - NWH::VehiclePhysics::CameraMouseDrag, [33](#)

- force
 - NWH::WheelController3D::WheelController::↔
Damper, [45](#)
 - NWH::WheelController3D::WheelController::↔
Friction, [71](#)
 - NWH::WheelController3D::WheelController::↔
Spring, [124](#)
 - NWH::WheelController3D::WheelController::↔
WheelHit, [207](#)
- forceCoefficient
 - NWH::WheelController3D::WheelController::↔
Friction, [71](#)
- forceCurve
 - NWH::WheelController3D::WheelController::↔
Spring, [124](#)
- forcedInduction
 - NWH::VehiclePhysics::Engine, [57](#)
- forward
 - NWH::WheelController3D::WheelController::↔
Wheel, [186](#)
- ForwardAcceleration
 - NWH::VehiclePhysics::VehicleController, [169](#)
- forwardDir
 - NWH::WheelController3D::WheelController::↔
WheelHit, [207](#)
- forwardFriction
 - NWH::WheelController3D::WheelController, [199](#)
- forwardFrictionCurve
 - NWH::WheelController3D::WheelController↔
Inspector, [205](#)
- ForwardGearCount
 - NWH::VehiclePhysics::Transmission, [151](#)
- ForwardGears
 - NWH::VehiclePhysics::Transmission, [151](#)
- forwardParams
 - NWH::WheelController3D::WheelController↔
Inspector, [205](#)
- ForwardScanResolution
 - NWH::WheelController3D::WheelController, [199](#)
- forwardSkidVolume
 - NWH::VehiclePhysics::SkidComponent, [104](#)
- ForwardSlip
 - NWH::VehiclePhysics::Wheel, [180](#)
- forwardSlip
 - NWH::WheelController3D::WheelController::↔
WheelHit, [207](#)
- ForwardSlipPercent
 - NWH::VehiclePhysics::Wheel, [180](#)
- forwardSlipThreshold
 - NWH::VehiclePhysics::VehicleController, [166](#)
- ForwardVelocity
 - NWH::VehiclePhysics::VehicleController, [169](#)
- freeRollingAngularVelocity
 - NWH::WheelController3D::WheelController::↔
Wheel, [186](#)
- FrictionPreset
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [74](#)
- FrictionPresetEnum
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [73](#)
- frictionPresetEnum
 - NWH::VehiclePhysics::GroundDetection::Ground↔
Entity, [85](#)
- FrictionPresetList
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [74](#)
- frictionTorque
 - NWH::VehiclePhysics::Brakes, [26](#)
- fuel
 - NWH::VehiclePhysics::VehicleController, [166](#)
- FuelCutoff
 - NWH::VehiclePhysics::Engine, [58](#)
- FuelPercentage
 - NWH::VehiclePhysics::Fuel, [78](#)
- fullBeamLights
 - NWH::VehiclePhysics::InputStates, [89](#)
- fullBeams
 - NWH::VehiclePhysics::Lights, [92](#)
- Gear
 - NWH::VehiclePhysics::Transmission, [151](#)
- gearChangeComponent
 - NWH::VehiclePhysics::Sound, [114](#)
- gearMultiplier
 - NWH::VehiclePhysics::Transmission, [148](#)
- GearName
 - NWH::VehiclePhysics::Transmission, [151](#)
- GearRatio
 - NWH::VehiclePhysics::Transmission, [152](#)
- Gears
 - NWH::VehiclePhysics::Transmission, [152](#)
- GenerateCamberCurve
 - NWH::WheelController3D::WheelController::↔
Wheel, [185](#)
- GenerateFrictionCurve
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [74](#)
- GenerateRimColliderMesh
 - NWH::WheelController3D::WheelController, [193](#)
- Generic
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [74](#)
- geometry
 - NWH::VehiclePhysics::Axle, [19](#)
- GetAverageWheelRPM
 - NWH::VehiclePhysics::VehicleController, [163](#)
- GetAxles
 - NWH::VehiclePhysics::VehicleController, [163](#)
- GetClampedValue
 - NWH::VehiclePhysics::SteeringWheel, [130](#)
- GetClutchEngagementAtPedalPosition
 - NWH::VehiclePhysics::Transmission, [145](#)
- GetCollisionState
 - NWH::VehiclePhysics::VehicleController, [163](#)
- GetCorrectWheelRpm
 - NWH::VehiclePhysics::VehicleController, [163](#)

- GetCurrentGroundEntity
 - NWH::VehiclePhysics::GroundDetection, [82](#)
- GetCurrentGroundEntityIndex
 - NWH::VehiclePhysics::GroundDetection, [83](#)
- GetCurrentSkidmarkTexture
 - NWH::VehiclePhysics::GroundDetection, [83](#)
- GetFrictionPreset
 - NWH::WheelController3D::WheelController, [193](#)
- GetGearRatio
 - NWH::VehiclePhysics::Transmission, [145](#)
- GetGroundHit
 - NWH::WheelController3D::WheelController, [193](#)
- GetMaxSpeedForGear
 - NWH::VehiclePhysics::Transmission, [145](#)
- GetMouseHorizontal
 - NWH::VehiclePhysics::VehicleController, [164](#)
- GetMouseVertical
 - NWH::VehiclePhysics::VehicleController, [164](#)
- GetRawAverageWheelRPM
 - NWH::VehiclePhysics::VehicleController, [164](#)
- GetWorldPose
 - NWH::WheelController3D::WheelController, [194](#)
- Grass
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [75](#)
- Gravel
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [75](#)
- groundDetection
 - NWH::VehiclePhysics::VehicleController, [166](#)
- groundEntities
 - NWH::VehiclePhysics::GroundDetection, [83](#)
- groundOffset
 - NWH::VehiclePhysics::SkidmarkGenerator, [107](#)
- groundPoint
 - NWH::WheelController3D::WheelController::↔
WheelHit, [208](#)
- Handbrake
 - NWH::VehiclePhysics::InputStates, [90](#)
- handbrakeCoefficient
 - NWH::VehiclePhysics::Axle, [20](#)
- HasForwardSlip
 - NWH::VehiclePhysics::Wheel, [181](#)
- HasFuel
 - NWH::VehiclePhysics::Fuel, [79](#)
- HasSideSlip
 - NWH::VehiclePhysics::Wheel, [181](#)
- hazardLights
 - NWH::VehiclePhysics::InputStates, [89](#)
- headLights
 - NWH::VehiclePhysics::Lights, [93](#)
- height
 - NWH::VehiclePhysics::CameraFollow, [30](#)
- highSpeedAngle
 - NWH::VehiclePhysics::Steering, [128](#)
- Horizontal
 - NWH::VehiclePhysics::InputStates, [90](#)
- horizontalMouseSensitivity
 - NWH::VehiclePhysics::CameraMouseDrag, [33](#)
- horn
 - NWH::VehiclePhysics::InputStates, [89](#)
- hornComponent
 - NWH::VehiclePhysics::Sound, [114](#)
- Ice
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [75](#)
- ignoreLayer
 - NWH::VehiclePhysics::VehicleFromScript, [172](#)
- ignoreTags
 - NWH::VehiclePhysics::DamageHandler, [44](#)
- inertia
 - NWH::WheelController3D::WheelController::↔
Wheel, [186](#)
- Initialize
 - NWH::VehiclePhysics::AirBrakeComponent, [16](#)
 - NWH::VehiclePhysics::Axle, [18](#)
 - NWH::VehiclePhysics::Backfire, [22](#)
 - NWH::VehiclePhysics::BackfireComponent, [23](#)
 - NWH::VehiclePhysics::BlinkerComponent, [24](#)
 - NWH::VehiclePhysics::CrashComponent, [40](#)
 - NWH::VehiclePhysics::DamageHandler, [42](#)
 - NWH::VehiclePhysics::DrivingAssists, [52](#)
 - NWH::VehiclePhysics::Effects, [53](#)
 - NWH::VehiclePhysics::Engine, [56](#)
 - NWH::VehiclePhysics::EngineIdleSoundComponent, [61](#)
 - NWH::VehiclePhysics::EngineStartStopComponent, [63](#)
 - NWH::VehiclePhysics::ExhaustSmoke, [64](#)
 - NWH::VehiclePhysics::FlipOver, [67](#)
 - NWH::VehiclePhysics::ForcedInduction, [69](#)
 - NWH::VehiclePhysics::Fuel, [77](#)
 - NWH::VehiclePhysics::GearChangeComponent, [79](#)
 - NWH::VehiclePhysics::HornComponent, [87](#)
 - NWH::VehiclePhysics::InputStates, [89](#)
 - NWH::VehiclePhysics::Lights, [92](#)
 - NWH::VehiclePhysics::Metrics, [96](#)
 - NWH::VehiclePhysics::SkidComponent, [103](#)
 - NWH::VehiclePhysics::SkidmarkGenerator, [106](#)
 - NWH::VehiclePhysics::Skidmarks, [108](#)
 - NWH::VehiclePhysics::Sound, [112](#)
 - NWH::VehiclePhysics::SoundComponent, [120](#)
 - NWH::VehiclePhysics::Steering, [127](#)
 - NWH::VehiclePhysics::SurfaceComponent, [132](#)
 - NWH::VehiclePhysics::SurfaceParticles, [133](#)
 - NWH::VehiclePhysics::SuspensionComponent, [134](#)
 - NWH::VehiclePhysics::Tracks, [135](#)
 - NWH::VehiclePhysics::TrailerHandler, [139](#)
 - NWH::VehiclePhysics::Transmission, [145](#)
 - NWH::VehiclePhysics::TransmissionWhine↔
Component, [153](#)
 - NWH::VehiclePhysics::TurboFlutterComponent, [155](#)

- NWH::VehiclePhysics::TurboWhistleComponent, 156
- NWH::VehiclePhysics::Wheel, 178
- NWH::WheelController3D::WheelController, 194
- NWH::WheelController3D::WheelController::↔ Wheel, 185
- input
 - NWH::VehiclePhysics::VehicleController, 166
- InputType
 - NWH::VehiclePhysics::DesktopInputManager, 47
 - NWH::VehiclePhysics::MobileInputManager, 98
- inputType
 - NWH::VehiclePhysics::DesktopInputManager, 47
 - NWH::VehiclePhysics::MobileInputManager, 100
- inside
 - NWH::WheelController3D::WheelController::↔ Wheel, 186
- insideVehicle
 - NWH::VehiclePhysics::Sound, 114
- intensity
 - NWH::VehiclePhysics::DrivingAssists::DrivingAid, 51
- intensityRange
 - NWH::VehiclePhysics::ExhaustSmoke, 65
- interiorAttenuation
 - NWH::VehiclePhysics::Sound, 114
- IsGrounded
 - NWH::VehiclePhysics::Wheel, 181
- isGrounded
 - NWH::WheelController3D::WheelController, 199
- IsPowered
 - NWH::VehiclePhysics::Axle, 20
- IsRunning
 - NWH::VehiclePhysics::Engine, 59
- isSolid
 - NWH::VehiclePhysics::Axle::Geometry, 81
- isTrailer
 - NWH::VehiclePhysics::TrailerHandler, 140
- lastShiftTime
 - NWH::VehiclePhysics::Transmission, 148
- LateUpdate
 - NWH::VehiclePhysics::VehicleController, 164
- leftBlinker
 - NWH::VehiclePhysics::InputStates, 89
- leftBlinkers
 - NWH::VehiclePhysics::Lights, 93
- leftSprockets
 - NWH::VehiclePhysics::Tracks, 136
- leftTrackRenderer
 - NWH::VehiclePhysics::Tracks, 136
- leftWheel
 - NWH::VehiclePhysics::Axle, 20
- length
 - NWH::WheelController3D::WheelController::↔ Spring, 124
- lifeDistance
 - NWH::VehiclePhysics::ExhaustSmoke, 65
- lightMeshRenderers
 - NWH::VehiclePhysics::Lights::VehicleLight, 174
- lightSources
 - NWH::VehiclePhysics::Lights::VehicleLight, 174
- lights
 - NWH::VehiclePhysics::Effects, 54
- Load
 - NWH::VehiclePhysics::VehicleController, 169
- Lockup
 - NWH::VehiclePhysics::Wheel, 178
- lowBeamLights
 - NWH::VehiclePhysics::InputStates, 89
- lowPassFrequency
 - NWH::VehiclePhysics::Sound, 114
- lowPassQ
 - NWH::VehiclePhysics::Sound, 115
- lowSpeedAngle
 - NWH::VehiclePhysics::Steering, 128
- markWidth
 - NWH::VehiclePhysics::SkidmarkGenerator, 107
- mass
 - NWH::WheelController3D::WheelController, 199
 - NWH::WheelController3D::WheelController::↔ Wheel, 186
- masterGroup
 - NWH::VehiclePhysics::Sound, 115
- masterVolume
 - NWH::VehiclePhysics::Sound, 115
- maxDetectionSpeed
 - NWH::VehiclePhysics::FlipOver, 68
- maxDistance
 - NWH::VehiclePhysics::CameraMouseDrag, 33
- maxDistortion
 - NWH::VehiclePhysics::EngineIdleSoundComponent, 62
- maxDownforceSpeed
 - NWH::VehiclePhysics::Downforce, 49
- maxEnterExitVehicleSpeed
 - NWH::VehiclePhysics::CharacterVehicleChanger, 38
- maxForce
 - NWH::VehiclePhysics::Downforce::Downforce↔ Point, 49
 - NWH::WheelController3D::WheelController::↔ Damper, 45
 - NWH::WheelController3D::WheelController::↔ Friction, 71
 - NWH::WheelController3D::WheelController::↔ Spring, 124
- maxLength
 - NWH::WheelController3D::WheelController::↔ Spring, 124
- maxMarks
 - NWH::VehiclePhysics::SkidmarkGenerator, 107
- maxMarksPerSection
 - NWH::VehiclePhysics::Skidmarks, 109
- maxNoTrailerPowerReduction
 - NWH::VehiclePhysics::TrailerHandler, 141
- maxPositionOffsetMagnitude

- NWH::VehiclePhysics::CameraOnboard, [35](#)
- maxPower
 - NWH::VehiclePhysics::Engine, [57](#)
- maxPowerGainMultiplier
 - NWH::VehiclePhysics::ForcedInduction, [70](#)
- MaxPutDownForce
 - NWH::WheelController3D::WheelController, [199](#)
- maxPutDownForce
 - NWH::WheelController3D::WheelController, [196](#)
- maxRPM
 - NWH::VehiclePhysics::Engine, [57](#)
- maxRpmChange
 - NWH::VehiclePhysics::Engine, [57](#)
- maxSkidmarkAlpha
 - NWH::VehiclePhysics::Skidmarks, [109](#)
- maxTorque
 - NWH::VehiclePhysics::Brakes, [26](#)
- maxTris
 - NWH::VehiclePhysics::SkidmarkGenerator, [107](#)
- maximumSteeringAngle
 - NWH::VehiclePhysics::SteeringWheel, [131](#)
- MeshDeform
 - NWH::VehiclePhysics::DamageHandler, [42](#)
- metrics
 - NWH::VehiclePhysics::VehicleController, [166](#)
- minDistance
 - NWH::VehiclePhysics::CameraMouseDrag, [33](#)
 - NWH::VehiclePhysics::Skidmarks, [109](#)
- minRPM
 - NWH::VehiclePhysics::Engine, [57](#)
- minSqrDistance
 - NWH::VehiclePhysics::SkidmarkGenerator, [107](#)
- MotorTorque
 - NWH::VehiclePhysics::Wheel, [181](#)
- motorTorque
 - NWH::WheelController3D::WheelController, [200](#)
 - NWH::WheelController3D::WheelController::←
Wheel, [186](#)
- mouseWheelSensitivity
 - NWH::VehiclePhysics::CameraMouseDrag, [33](#)
- NWH.VehiclePhysics, [11](#)
- NWH.VehiclePhysics.ABS, [15](#)
- NWH.VehiclePhysics.AirBrakeComponent, [16](#)
- NWH.VehiclePhysics.Axle, [17](#)
- NWH.VehiclePhysics.Axle.Geometry, [80](#)
- NWH.VehiclePhysics.Backfire, [21](#)
- NWH.VehiclePhysics.BackfireComponent, [23](#)
- NWH.VehiclePhysics.BlinkerComponent, [24](#)
- NWH.VehiclePhysics.Brakes, [25](#)
- NWH.VehiclePhysics.CameraChanger, [27](#)
- NWH.VehiclePhysics.CameraFollow, [29](#)
- NWH.VehiclePhysics.CameraInsideVehicle, [31](#)
- NWH.VehiclePhysics.CameraMouseDrag, [31](#)
- NWH.VehiclePhysics.CameraOnboard, [34](#)
- NWH.VehiclePhysics.CenterOfMass, [36](#)
- NWH.VehiclePhysics.CharacterVehicleChanger, [37](#)
- NWH.VehiclePhysics.CrashComponent, [39](#)
- NWH.VehiclePhysics.DamageHandler, [41](#)
- NWH.VehiclePhysics.DamageHandler.CollisionEvent, [38](#)
- NWH.VehiclePhysics.DesktopInputManager, [46](#)
- NWH.VehiclePhysics.Downforce, [48](#)
- NWH.VehiclePhysics.Downforce.DownforcePoint, [49](#)
- NWH.VehiclePhysics.DrivingAssists, [51](#)
- NWH.VehiclePhysics.DrivingAssists.DrivingAid, [50](#)
- NWH.VehiclePhysics.Effects, [53](#)
- NWH.VehiclePhysics.Engine, [54](#)
- NWH.VehiclePhysics.EngineIdleSoundComponent, [61](#)
- NWH.VehiclePhysics.EngineStartStopComponent, [63](#)
- NWH.VehiclePhysics.ExhaustSmoke, [64](#)
- NWH.VehiclePhysics.FlipOver, [66](#)
- NWH.VehiclePhysics.ForcedInduction, [68](#)
- NWH.VehiclePhysics.Fuel, [76](#)
- NWH.VehiclePhysics.GearChangeComponent, [79](#)
- NWH.VehiclePhysics.GroundDetection, [82](#)
- NWH.VehiclePhysics.GroundDetection.GroundEntity, [84](#)
- NWH.VehiclePhysics.GroundDetection.SoundComponent, [117](#)
- NWH.VehiclePhysics.HornComponent, [87](#)
- NWH.VehiclePhysics.InputStates, [88](#)
- NWH.VehiclePhysics.Lights, [91](#)
- NWH.VehiclePhysics.Lights.VehicleLight, [173](#)
- NWH.VehiclePhysics.Metrics, [95](#)
- NWH.VehiclePhysics.Metrics.Metric, [94](#)
- NWH.VehiclePhysics.MobileInputManager, [97](#)
- NWH.VehiclePhysics.RenderCameraDisable, [101](#)
- NWH.VehiclePhysics.Rigging, [101](#)
- NWH.VehiclePhysics.SkidComponent, [103](#)
- NWH.VehiclePhysics.SkidmarkDestroy, [104](#)
- NWH.VehiclePhysics.SkidmarkGenerator, [105](#)
- NWH.VehiclePhysics.Skidmarks, [108](#)
- NWH.VehiclePhysics.Sound, [110](#)
- NWH.VehiclePhysics.SoundComponent, [118](#)
- NWH.VehiclePhysics.Stability, [126](#)
- NWH.VehiclePhysics.Steering, [126](#)
- NWH.VehiclePhysics.SteeringWheel, [129](#)
- NWH.VehiclePhysics.SurfaceComponent, [131](#)
- NWH.VehiclePhysics.SurfaceParticles, [132](#)
- NWH.VehiclePhysics.SuspensionComponent, [134](#)
- NWH.VehiclePhysics.Tracks, [135](#)
- NWH.VehiclePhysics.TractionControl, [137](#)
- NWH.VehiclePhysics.TrailerHandler, [138](#)
- NWH.VehiclePhysics.Transmission, [142](#)
- NWH.VehiclePhysics.TransmissionWhineComponent, [153](#)
- NWH.VehiclePhysics.TurboFlutterComponent, [154](#)
- NWH.VehiclePhysics.TurboWhistleComponent, [155](#)
- NWH.VehiclePhysics.VehicleChanger, [157](#)
- NWH.VehiclePhysics.VehicleController, [160](#)
- NWH.VehiclePhysics.VehicleControllerInspector, [171](#)
- NWH.VehiclePhysics.VehicleFromScript, [171](#)
- NWH.VehiclePhysics.Wheel, [175](#)
- NWH.WheelController3D.WheelController, [189](#)
- NWH.WheelController3D.WheelController.Damper, [45](#)
- NWH.WheelController3D.WheelController.Friction, [71](#)

- NWH.WheelController3D.WheelController.Friction↔
 - Preset, 72
- NWH.WheelController3D.WheelController.Spring, 123
- NWH.WheelController3D.WheelController.Wheel, 184
- NWH.WheelController3D.WheelController.WheelHit, 205
- NWH.WheelController3D.WheelControllerInspector, 204
- NWH.WheelController3D, 14
- NWH::VehiclePhysics::ABS
 - Update, 15
- NWH::VehiclePhysics::AirBrakeComponent
 - Initialize, 16
 - Update, 16
- NWH::VehiclePhysics::Axle
 - Bias, 20
 - brakeCoefficient, 19
 - differentialStrength, 19
 - DifferentialType, 18
 - differentialType, 19
 - geometry, 19
 - handbrakeCoefficient, 20
 - Initialize, 18
 - IsPowered, 20
 - leftWheel, 20
 - NoSlipRPM, 21
 - powerCoefficient, 20
 - RPM, 21
 - rightWheel, 20
 - SmoothRPM, 21
 - TorqueSplit, 18
 - Update, 19
 - WheelSpin, 21
- NWH::VehiclePhysics::Axle::Geometry
 - ackermannPercent, 80
 - antiRollBarForce, 81
 - camberAtBottom, 81
 - camberAtTop, 81
 - casterAngle, 81
 - isSolid, 81
 - steerCoefficient, 81
 - toeAngle, 81
- NWH::VehiclePhysics::Backfire
 - duration, 22
 - Flash, 22
 - Initialize, 22
 - renderers, 22
 - Update, 22
- NWH::VehiclePhysics::BackfireComponent
 - Initialize, 23
 - Update, 23
- NWH::VehiclePhysics::BlinkerComponent
 - Initialize, 24
 - Update, 24
- NWH::VehiclePhysics::Brakes
 - Active, 27
 - airBrakePressure, 26
 - airBrakes, 26
 - brakeWhileIdle, 26
 - frictionTorque, 26
 - maxTorque, 26
 - smoothing, 26
 - Update, 25
- NWH::VehiclePhysics::CameraChanger
 - cameraTag, 28
 - currentCamera, 28
 - NextCamera, 28
 - vehicleCameras, 28
- NWH::VehiclePhysics::CameraFollow
 - angleFollowStrength, 30
 - AngleSigned, 29
 - distance, 30
 - height, 30
 - smoothing, 30
 - target, 30
 - targetForwardOffset, 30
 - targetUpOffset, 31
- NWH::VehiclePhysics::CameraMouseDrag
 - ClampAngle, 32
 - distance, 32
 - distanceSmoothing, 32
 - followTargetsRotation, 33
 - horizontalMouseSensitivity, 33
 - maxDistance, 33
 - minDistance, 33
 - mouseWheelSensitivity, 33
 - smoothing, 33
 - target, 33
 - verticalMaxAngle, 34
 - verticalMinAngle, 34
 - verticalMouseSensitivity, 34
- NWH::VehiclePhysics::CameraOnboard
 - maxPositionOffsetMagnitude, 35
 - positionIntensity, 35
 - positionSmoothing, 35
 - vehicleController, 35
- NWH::VehiclePhysics::CenterOfMass
 - centerOfMassOffset, 36
 - showCOM, 36
- NWH::VehiclePhysics::CharacterVehicleChanger
 - characterControllerObject, 37
 - enterDistance, 37
 - enterExitTag, 37
 - maxEnterExitVehicleSpeed, 38
 - nearVehicle, 38
- NWH::VehiclePhysics::CrashComponent
 - Initialize, 40
 - Play, 40
 - Update, 40
- NWH::VehiclePhysics::DamageHandler
 - allowedDamage, 42
 - Damage, 44
 - DamagePercent, 44
 - decelerationThreshold, 43
 - deformationRadius, 43
 - deformationRandomness, 43

- deformationStrength, 43
- deformationVerticesPerFrame, 43
- enabled, 43
- Enqueue, 42
- ignoreTags, 44
- Initialize, 42
- MeshDeform, 42
- performanceDegradation, 44
- previousCollisionHash, 44
- Update, 42
- NWH::VehiclePhysics::DamageHandler::CollisionEvent
 - collision, 39
 - decelerationMagnitude, 39
 - deformationQueue, 39
- NWH::VehiclePhysics::DesktopInputManager
 - InputType, 47
 - inputType, 47
 - vehicleChanger, 47
 - vehicleController, 47
- NWH::VehiclePhysics::Downforce
 - downforcePoints, 48
 - maxDownforceSpeed, 49
- NWH::VehiclePhysics::Downforce::DownforcePoint
 - maxForce, 49
 - position, 49
- NWH::VehiclePhysics::DrivingAssists
 - abs, 52
 - Initialize, 52
 - stability, 52
 - tcs, 52
 - Update, 52
- NWH::VehiclePhysics::DrivingAssists::DrivingAid
 - active, 50
 - enabled, 50
 - intensity, 51
- NWH::VehiclePhysics::Effects
 - exhaustFlash, 53
 - exhausts, 53
 - Initialize, 53
 - lights, 54
 - skidmarks, 54
 - Update, 53
- NWH::VehiclePhysics::Engine
 - forcedInduction, 57
 - FuelCutoff, 58
 - Initialize, 56
 - IsRunning, 59
 - maxPower, 57
 - maxRPM, 57
 - maxRpmChange, 57
 - minRPM, 57
 - Power, 59
 - powerCurve, 57
 - PowerInHP, 59
 - RPMPercent, 59
 - RPM, 59
 - RpmOverflow, 59
 - runOnStartup, 58
 - Start, 56
 - starting, 58
 - StartingPercent, 60
 - Stop, 56
 - stopping, 58
 - StoppingPercent, 60
 - TcsPowerReduction, 60
 - throttleSmoothing, 58
 - Toggle, 56
 - Torque, 60
 - TotalPowerReduction, 60
 - Update, 56
- NWH::VehiclePhysics::EngineIdleSoundComponent
 - Initialize, 61
 - maxDistortion, 62
 - pitchRange, 62
 - smoothing, 62
 - Update, 61
 - volumeRange, 62
- NWH::VehiclePhysics::EngineStartStopComponent
 - Initialize, 63
 - Update, 63
- NWH::VehiclePhysics::ExhaustSmoke
 - baseIntensity, 65
 - emit, 65
 - Initialize, 64
 - intensityRange, 65
 - lifeDistance, 65
 - particleSystems, 65
 - soot, 65
 - sootColor, 66
 - startSize, 66
 - Update, 64
 - vaporColor, 66
- NWH::VehiclePhysics::FlipOver
 - allowedAngle, 67
 - enabled, 67
 - flippedOver, 67
 - Initialize, 67
 - maxDetectionSpeed, 68
 - rotationSpeed, 68
 - timeout, 68
 - Update, 67
- NWH::VehiclePhysics::ForcedInduction
 - flutterSoundFlag, 69
 - Initialize, 69
 - maxPowerGainMultiplier, 70
 - PowerGainMultiplier, 70
 - SpoolPercent, 70
 - spoolUpTime, 70
 - Update, 69
 - useForcedInduction, 70
- NWH::VehiclePhysics::Fuel
 - amount, 77
 - capacity, 77
 - ConsumptionKilometersPerLiter, 78
 - ConsumptionLitersPer100Kilometers, 78
 - ConsumptionLitersPerSecond, 78

- ConsumptionMPG, 78
- efficiency, 77
- FuelPercentage, 78
- HasFuel, 79
- Initialize, 77
- Update, 77
- useFuel, 78
- NWH::VehiclePhysics::GearChangeComponent
 - Initialize, 79
 - Update, 80
- NWH::VehiclePhysics::GroundDetection
 - dustPrefab, 83
 - GetCurrentGroundEntity, 82
 - GetCurrentGroundEntityIndex, 83
 - GetCurrentSkidmarkTexture, 83
 - groundEntities, 83
 - smokePrefab, 83
- NWH::VehiclePhysics::GroundDetection::GroundEntity
 - dustColor, 85
 - dustIntensity, 85
 - frictionPresetEnum, 85
 - name, 85
 - skidSoundComponent, 85
 - skidmarkMaterial, 85
 - slipBasedSkidIntensity, 86
 - slipSensitiveSurfaceSound, 86
 - smokeIntensity, 86
 - surfaceSoundComponent, 86
 - tags, 86
 - terrainTextureIndices, 86
- NWH::VehiclePhysics::GroundDetection::Sound←Component
 - clip, 117
 - pitch, 117
 - source, 118
 - volume, 118
- NWH::VehiclePhysics::HornComponent
 - Initialize, 87
 - Update, 87
- NWH::VehiclePhysics::InputStates
 - Clutch, 90
 - fullBeamLights, 89
 - Handbrake, 90
 - hazardLights, 89
 - Horizontal, 90
 - horn, 89
 - Initialize, 89
 - leftBlinker, 89
 - lowBeamLights, 89
 - RawVertical, 90
 - rightBlinker, 89
 - ShiftDown, 90
 - ShiftUp, 90
 - trailerAttachDetach, 90
 - Vertical, 91
- NWH::VehiclePhysics::Lights
 - BlinkerState, 94
 - enabled, 92
 - fullBeams, 92
 - headLights, 93
 - Initialize, 92
 - leftBlinkers, 93
 - rearLights, 93
 - rightBlinkers, 93
 - stopLights, 93
 - TurnOffAllLights, 92
 - Update, 92
- NWH::VehiclePhysics::Lights::VehicleLight
 - active, 174
 - lightMeshRenderers, 174
 - lightSources, 174
 - On, 174
 - TurnOff, 173
 - TurnOn, 174
- NWH::VehiclePhysics::Metrics
 - averageSpeed, 96
 - continousDriftDistance, 96
 - continousDriftTime, 96
 - Initialize, 96
 - odometer, 96
 - topSpeed, 96
 - totalDriftDistance, 97
 - totalDriftTime, 97
 - Update, 96
- NWH::VehiclePhysics::Metrics::Metric
 - Reset, 94
 - Update, 94
 - UpdateDelegate, 94
 - value, 95
- NWH::VehiclePhysics::MobileInputManager
 - BrakeDown, 98
 - BrakeUp, 98
 - ChangeCamera, 99
 - ChangeVehicle, 99
 - EngineStartStop, 99
 - InputType, 98
 - inputType, 100
 - steeringWheel, 100
 - ThrottleDown, 99
 - ThrottleUp, 99
 - tiltSensitivity, 100
 - ToggleHighBeams, 99
 - ToggleLowBeams, 99
 - TrailerAttachDetach, 99
 - UpdateSteering, 100
 - vehicleChanger, 100
 - vehicleController, 100
- NWH::VehiclePhysics::Rigging
 - axleBones, 102
 - enabled, 102
 - solidAxle, 102
 - Update, 102
 - wheelBones, 102
- NWH::VehiclePhysics::SkidComponent
 - forwardSkidVolume, 104
 - Initialize, 103

- sideSkidVolume, [104](#)
- Update, [103](#)
- NWH::VehiclePhysics::SkidmarkDestroy
 - distanceThreshold, [105](#)
 - parentVehicleController, [105](#)
- NWH::VehiclePhysics::SkidmarkGenerator
 - CreateNewSnapshot, [106](#)
 - DoubleSubArray, [106](#)
 - groundOffset, [107](#)
 - Initialize, [106](#)
 - markWidth, [107](#)
 - maxMarks, [107](#)
 - maxTris, [107](#)
 - minSqrDistance, [107](#)
 - SubArray, [106](#)
 - Update, [106](#)
 - wheel, [107](#)
- NWH::VehiclePhysics::Skidmarks
 - enabled, [108](#)
 - Initialize, [108](#)
 - maxMarksPerSection, [109](#)
 - maxSkidmarkAlpha, [109](#)
 - minDistance, [109](#)
 - persistentSkidmarkDistance, [109](#)
 - persistentSkidmarks, [109](#)
 - skidmarkStrength, [109](#)
 - Update, [108](#)
- NWH::VehiclePhysics::Sound
 - airBrakeComponent, [113](#)
 - blinkerComponent, [113](#)
 - crashComponent, [113](#)
 - crashMixerGroup, [113](#)
 - Disable, [111](#)
 - Enable, [111](#)
 - engineIdleComponent, [113](#)
 - engineMixerGroup, [113](#)
 - engineStartStopComponent, [113](#)
 - exhaustPopComponent, [114](#)
 - gearChangeComponent, [114](#)
 - hornComponent, [114](#)
 - Initialize, [112](#)
 - insideVehicle, [114](#)
 - interiorAttenuation, [114](#)
 - lowPassFrequency, [114](#)
 - lowPassQ, [115](#)
 - masterGroup, [115](#)
 - masterVolume, [115](#)
 - otherMixerGroup, [115](#)
 - SetAudioSourceDefaults, [112](#)
 - SetDefaults, [112](#)
 - skidComponent, [115](#)
 - spatialBlend, [115](#)
 - surfaceComponent, [115](#)
 - surfaceNoiseMixerGroup, [116](#)
 - suspensionComponent, [116](#)
 - suspensionMixerGroup, [116](#)
 - transmissionMixerGroup, [116](#)
 - transmissionWhineComponent, [116](#)
 - turboFlutterComponent, [116](#)
 - turboMixerGroup, [116](#)
 - turboWhistleComponent, [117](#)
 - Update, [112](#)
- NWH::VehiclePhysics::SoundComponent
 - audioMixerGroup, [121](#)
 - Clip, [122](#)
 - Clips, [122](#)
 - clips, [121](#)
 - Disable, [120](#)
 - Enable, [120](#)
 - Initialize, [120](#)
 - pitch, [121](#)
 - RandomClip, [122](#)
 - RegisterSources, [120](#)
 - SetVolume, [120](#)
 - Source, [122](#)
 - Sources, [123](#)
 - sources, [121](#)
 - Update, [121](#)
 - vc, [122](#)
 - volume, [122](#)
- NWH::VehiclePhysics::Stability
 - Update, [126](#)
- NWH::VehiclePhysics::Steering
 - AdjustGeometry, [127](#)
 - Angle, [129](#)
 - crossoverSpeed, [128](#)
 - degreesPerSecondLimit, [128](#)
 - highSpeedAngle, [128](#)
 - Initialize, [127](#)
 - lowSpeedAngle, [128](#)
 - Steer, [127](#)
 - steeringWheel, [128](#)
 - steeringWheelTurnRatio, [128](#)
 - trackedSteerIntensity, [129](#)
- NWH::VehiclePhysics::SteeringWheel
 - DragEvent, [130](#)
 - GetClampedValue, [130](#)
 - maximumSteeringAngle, [131](#)
 - PressEvent, [130](#)
 - ReleaseEvent, [130](#)
 - returnToCenterSpeed, [131](#)
 - steeringWheelGraphic, [131](#)
- NWH::VehiclePhysics::SurfaceComponent
 - Initialize, [132](#)
 - Update, [132](#)
- NWH::VehiclePhysics::SurfaceParticles
 - dust, [133](#)
 - dustGo, [133](#)
 - Initialize, [133](#)
 - smoke, [133](#)
 - smokeGo, [133](#)
 - Update, [133](#)
- NWH::VehiclePhysics::SuspensionComponent
 - Initialize, [134](#)
 - Update, [134](#)
- NWH::VehiclePhysics::Tracks

- Initialize, 135
- leftSprockets, 136
- leftTrackRenderer, 136
- rightSprockets, 136
- rightTrackRenderer, 136
- textureOffsetSpeedCoefficient, 136
- trackedVehicle, 137
- Update, 136
- NWH::VehiclePhysics::TractionControl
 - powerReduction, 138
 - prevPowerReduction, 138
 - Update, 138
- NWH::VehiclePhysics::TrailerHandler
 - attachDistanceThreshold, 140
 - attachOnPlay, 140
 - attached, 140
 - attachmentPoint, 140
 - breakForce, 140
 - Initialize, 139
 - isTrailer, 140
 - maxNoTrailerPowerReduction, 141
 - NoTrailerPowerReduction, 141
 - trailerInRange, 141
 - trailerStand, 141
 - trailersTag, 141
 - Update, 139
- NWH::VehiclePhysics::Transmission
 - AddedClutchRPM, 150
 - AdjustedShiftDownRpm, 150
 - AdjustedShiftUpRpm, 150
 - automaticClutch, 147
 - CanShift, 151
 - clutchEngagementCurve, 147
 - clutchPedalPressedPercent, 148
 - ClutchPercent, 151
 - DifferentialType, 144
 - differentialType, 148
 - dynamicShiftPoint, 148
 - ForwardGearCount, 151
 - ForwardGears, 151
 - Gear, 151
 - gearMultiplier, 148
 - GearName, 151
 - GearRatio, 152
 - Gears, 152
 - GetClutchEngagementAtPedalPosition, 145
 - GetGearRatio, 145
 - GetMaxSpeedForGear, 145
 - Initialize, 145
 - lastShiftTime, 148
 - postShiftBan, 149
 - RPM, 152
 - ReconstructGearList, 146
 - ReverseGearCount, 152
 - ReverseGears, 152
 - ReverseRPM, 152
 - ReverseTransmitRPM, 146
 - ReverseTransmitTorque, 146
 - shiftDuration, 149
 - shiftDurationRandomness, 149
 - ShiftInto, 146
 - shiftPointRandomness, 149
 - Shifting, 153
 - targetClutchRPM, 149
 - targetShiftDownRPM, 149
 - targetShiftUpRPM, 150
 - TorqueSplit, 146
 - TransmissionType, 144
 - transmissionType, 150
 - TransmitRPM, 147
 - TransmitTorque, 147
 - Update, 147
 - UpdateClutch, 147
- NWH::VehiclePhysics::TransmissionWhineComponent
 - Initialize, 153
 - pitchRange, 154
 - Update, 154
- NWH::VehiclePhysics::TurboFlutterComponent
 - Initialize, 155
 - Update, 155
- NWH::VehiclePhysics::TurboWhistleComponent
 - Initialize, 156
 - pitchRange, 156
 - Update, 156
- NWH::VehiclePhysics::VehicleChanger
 - ActiveVehicleController, 159
 - ChangeVehicle, 157, 158
 - CharacterBased, 159
 - CurrentVehicle, 159
 - DeactivateAll, 159
 - NearestVehicleFrom, 158
 - NextVehicle, 158
 - vehicleTag, 158
 - vehicles, 158
- NWH::VehiclePhysics::VehicleController
 - Acceleration, 168
 - Active, 168
 - AngleSigned, 162
 - axles, 165
 - brakes, 165
 - CollisionInfo, 168
 - CollisionState, 168
 - damage, 165
 - DetectWheelSkid, 163
 - DetectWheelSpin, 163
 - Direction, 168
 - drivingAids, 165
 - effects, 166
 - engine, 166
 - flipOver, 166
 - ForwardAcceleration, 169
 - forwardSlipThreshold, 166
 - ForwardVelocity, 169
 - fuel, 166
 - GetAverageWheelIRPM, 163
 - GetAxles, 163

- GetCollisionState, [163](#)
- GetCorrectWheelRpm, [163](#)
- GetMouseHorizontal, [164](#)
- GetMouseVertical, [164](#)
- GetRawAverageWheelRPM, [164](#)
- groundDetection, [166](#)
- input, [166](#)
- LateUpdate, [164](#)
- Load, [169](#)
- metrics, [166](#)
- Reset, [164](#)
- ResetCOM, [164](#)
- rigging, [167](#)
- SetDefaults, [165](#)
- sideSlipThreshold, [167](#)
- sound, [167](#)
- Speed, [169](#)
- SpeedKPH, [169](#)
- speedLimiter, [167](#)
- SpeedMPH, [169](#)
- steering, [167](#)
- tracks, [167](#)
- trailer, [167](#)
- transmission, [167](#)
- Update, [165](#)
- VehicleCollisionState, [162](#)
- vehicleRigidbody, [168](#)
- WheelSkid, [170](#)
- WheelSpeed, [170](#)
- WheelSpin, [170](#)
- Wheels, [170](#)
- NWH::VehiclePhysics::VehicleControllerInspector
 - OnInspectorGUI, [171](#)
- NWH::VehiclePhysics::VehicleFromScript
 - ignoreLayer, [172](#)
 - vehicleMass, [172](#)
 - wheelRadius, [172](#)
 - wheelTag, [172](#)
 - wheelWidth, [172](#)
- NWH::VehiclePhysics::Wheel
 - Activate, [177](#)
 - AddBrakeTorque, [177](#)
 - Bias, [179](#)
 - brakeCoefficient, [179](#)
 - BrakeTorque, [179](#)
 - ControllerGO, [179](#)
 - ControllerTransform, [179](#)
 - CurrentGroundEntity, [180](#)
 - CurrentGroundEntityName, [180](#)
 - Damage, [180](#)
 - DamageSteerDirection, [180](#)
 - ForwardSlip, [180](#)
 - ForwardSlipPercent, [180](#)
 - HasForwardSlip, [181](#)
 - HasSideSlip, [181](#)
 - Initialize, [178](#)
 - IsGrounded, [181](#)
 - Lockup, [178](#)
 - MotorTorque, [181](#)
 - NoSlipRPM, [181](#)
 - RPM, [182](#)
 - Radius, [181](#)
 - ResetBrakes, [178](#)
 - SetBrakeIntensity, [178](#)
 - SideSlip, [182](#)
 - SideSlipPercent, [182](#)
 - SmoothForwardSlip, [182](#)
 - SmoothRPM, [182](#)
 - SmoothSideSlip, [182](#)
 - SpringTravel, [183](#)
 - SteerAngle, [183](#)
 - Suspend, [178](#)
 - Update, [178](#)
 - VisualTransform, [183](#)
 - Wheel, [177](#)
 - WheelController, [183](#)
 - wheelController, [179](#)
 - Width, [183](#)
- NWH::WheelController3D::WheelController
 - activeFrictionPreset, [195](#)
 - activeFrictionPresetEnum, [195](#)
 - angularVelocity, [198](#)
 - applyForceToOthers, [196](#)
 - brakeTorque, [198](#)
 - camber, [198](#)
 - damper, [196](#)
 - damperCurve, [198](#)
 - damperForce, [198](#)
 - damperUnitBumpForce, [198](#)
 - damperUnitReboundForce, [199](#)
 - debug, [196](#)
 - DetermineSide, [193](#)
 - fFriction, [196](#)
 - FixedUpdate, [193](#)
 - forwardFriction, [199](#)
 - ForwardScanResolution, [199](#)
 - GenerateRimColliderMesh, [193](#)
 - GetFrictionPreset, [193](#)
 - GetGroundHit, [193](#)
 - GetWorldPose, [194](#)
 - Initialize, [194](#)
 - isGrounded, [199](#)
 - mass, [199](#)
 - MaxPutDownForce, [199](#)
 - maxPutDownForce, [196](#)
 - motorTorque, [200](#)
 - NonRotating, [200](#)
 - Parent, [200](#)
 - parent, [196](#)
 - pointVelocity, [200](#)
 - radius, [200](#)
 - rimOffset, [200](#)
 - rpm, [201](#)
 - sFriction, [196](#)
 - ScanIgnoreLayers, [201](#)
 - SetActiveFrictionPreset, [194](#)

- SetCamber, [194](#), [195](#)
- Side, [192](#)
- sideFriction, [201](#)
- SideToSideScanResolution, [201](#)
- singleRay, [197](#)
- singleWheelHit, [197](#)
- speed, [201](#)
- spring, [197](#)
- springBottomedOut, [201](#)
- springCompression, [202](#)
- springCurve, [202](#)
- springLength, [202](#)
- springMaximumForce, [202](#)
- springOverExtended, [202](#)
- springTravel, [202](#)
- springTravelPoint, [203](#)
- springVelocity, [203](#)
- Start, [195](#)
- steerAngle, [203](#)
- suspensionForce, [203](#)
- tireRadius, [203](#)
- tireWidth, [203](#)
- trackedVehicle, [197](#)
- useRimCollider, [197](#)
- VehicleSide, [204](#)
- Visual, [204](#)
- wheel, [197](#)
- wheelHit, [197](#)
- NWH::WheelController3D::WheelController::Damper
 - dampingCurve, [45](#)
 - force, [45](#)
 - maxForce, [45](#)
 - unitBumpForce, [46](#)
 - unitReboundForce, [46](#)
- NWH::WheelController3D::WheelController::Friction
 - force, [71](#)
 - forceCoefficient, [71](#)
 - maxForce, [71](#)
 - slip, [72](#)
 - slipCoefficient, [72](#)
 - speed, [72](#)
- NWH::WheelController3D::WheelController::Friction↔
 - Preset
 - Arcade, [74](#)
 - BCDE, [74](#)
 - Curve, [76](#)
 - FrictionPreset, [74](#)
 - FrictionPresetEnum, [73](#)
 - FrictionPresetList, [74](#)
 - GenerateFrictionCurve, [74](#)
 - Generic, [74](#)
 - Grass, [75](#)
 - Gravel, [75](#)
 - Ice, [75](#)
 - name, [75](#)
 - Sand, [75](#)
 - Snow, [75](#)
 - TarmacDry, [75](#)
 - TarmacWet, [76](#)
 - Tracks, [76](#)
 - NWH::WheelController3D::WheelController::Spring
 - bottomOutForce, [124](#)
 - bottomedOut, [123](#)
 - compressionPercent, [124](#)
 - force, [124](#)
 - forceCurve, [124](#)
 - length, [124](#)
 - maxForce, [124](#)
 - maxLength, [124](#)
 - overExtended, [124](#)
 - overflow, [125](#)
 - overflowVelocity, [125](#)
 - prevLength, [125](#)
 - prevOverflow, [125](#)
 - targetPoint, [125](#)
 - velocity, [125](#)
 - NWH::WheelController3D::WheelController::Wheel
 - acceleration, [185](#)
 - angularVelocity, [185](#)
 - brakeTorque, [185](#)
 - camberAngle, [185](#)
 - camberCurve, [185](#)
 - forward, [186](#)
 - freeRollingAngularVelocity, [186](#)
 - GenerateCamberCurve, [185](#)
 - inertia, [186](#)
 - Initialize, [185](#)
 - inside, [186](#)
 - mass, [186](#)
 - motorTorque, [186](#)
 - nonRotating, [186](#)
 - nonRotatingPostionOffset, [186](#)
 - prevGroundPoint, [187](#)
 - prevVelocity, [187](#)
 - prevWorldPosition, [187](#)
 - residualAngularVelocity, [187](#)
 - right, [187](#)
 - rim, [187](#)
 - rimCollider, [187](#)
 - rimOffset, [187](#)
 - rotationAngle, [188](#)
 - rpm, [188](#)
 - steerAngle, [188](#)
 - tireLoad, [188](#)
 - tireRadius, [188](#)
 - up, [188](#)
 - velocity, [188](#)
 - visual, [188](#)
 - width, [189](#)
 - worldPosition, [189](#)
 - worldRotation, [189](#)
 - NWH::WheelController3D::WheelController::WheelHit
 - angleForward, [207](#)
 - collider, [209](#)
 - Copy, [207](#)
 - curvatureOffset, [207](#)

- distanceFromTire, [207](#)
- force, [207](#)
- forwardDir, [207](#)
- forwardSlip, [207](#)
- groundPoint, [208](#)
- normal, [209](#)
- offset, [208](#)
- point, [209](#)
- raycastHit, [208](#)
- sidewaysDir, [208](#)
- sidewaysSlip, [208](#)
- valid, [208](#)
- weight, [208](#)
- WheelHit, [206](#)
- NWH::WheelController3D::WheelControllerInspector
 - forwardFrictionCurve, [205](#)
 - forwardParams, [205](#)
 - OnInspectorGUI, [205](#)
 - sideFrictionCurve, [205](#)
 - sideParams, [205](#)
- NWH, [11](#)
- name
 - NWH::VehiclePhysics::GroundDetection::Ground↔Entity, [85](#)
 - NWH::WheelController3D::WheelController::↔FrictionPreset, [75](#)
- nearVehicle
 - NWH::VehiclePhysics::CharacterVehicleChanger, [38](#)
- NearestVehicleFrom
 - NWH::VehiclePhysics::VehicleChanger, [158](#)
- NextCamera
 - NWH::VehiclePhysics::CameraChanger, [28](#)
- NextVehicle
 - NWH::VehiclePhysics::VehicleChanger, [158](#)
- NoSlipRPM
 - NWH::VehiclePhysics::Axle, [21](#)
 - NWH::VehiclePhysics::Wheel, [181](#)
- NoTrailerPowerReduction
 - NWH::VehiclePhysics::TrailerHandler, [141](#)
- NonRotating
 - NWH::WheelController3D::WheelController, [200](#)
- nonRotating
 - NWH::WheelController3D::WheelController::↔Wheel, [186](#)
- nonRotatingPostionOffset
 - NWH::WheelController3D::WheelController::↔Wheel, [186](#)
- normal
 - NWH::WheelController3D::WheelController::↔WheelHit, [209](#)
- odometer
 - NWH::VehiclePhysics::Metrics, [96](#)
- offset
 - NWH::WheelController3D::WheelController::↔WheelHit, [208](#)
- On
 - NWH::VehiclePhysics::Lights::VehicleLight, [174](#)
- OnInspectorGUI
 - NWH::VehiclePhysics::VehicleControllerInspector, [171](#)
 - NWH::WheelController3D::WheelController↔Inspector, [205](#)
- otherMixerGroup
 - NWH::VehiclePhysics::Sound, [115](#)
- overExtended
 - NWH::WheelController3D::WheelController::↔Spring, [124](#)
- overflow
 - NWH::WheelController3D::WheelController::↔Spring, [125](#)
- overflowVelocity
 - NWH::WheelController3D::WheelController::↔Spring, [125](#)
- Parent
 - NWH::WheelController3D::WheelController, [200](#)
- parent
 - NWH::WheelController3D::WheelController, [196](#)
- parentVehicleController
 - NWH::VehiclePhysics::SkidmarkDestroy, [105](#)
- particleSystems
 - NWH::VehiclePhysics::ExhaustSmoke, [65](#)
- performanceDegradation
 - NWH::VehiclePhysics::DamageHandler, [44](#)
- persistentSkidmarkDistance
 - NWH::VehiclePhysics::Skidmarks, [109](#)
- persistentSkidmarks
 - NWH::VehiclePhysics::Skidmarks, [109](#)
- pitch
 - NWH::VehiclePhysics::GroundDetection::Sound↔Component, [117](#)
 - NWH::VehiclePhysics::SoundComponent, [121](#)
- pitchRange
 - NWH::VehiclePhysics::EngineIdleSoundComponent, [62](#)
 - NWH::VehiclePhysics::TransmissionWhine↔Component, [154](#)
 - NWH::VehiclePhysics::TurboWhistleComponent, [156](#)
- Play
 - NWH::VehiclePhysics::CrashComponent, [40](#)
- point
 - NWH::WheelController3D::WheelController::↔WheelHit, [209](#)
- pointVelocity
 - NWH::WheelController3D::WheelController, [200](#)
- position
 - NWH::VehiclePhysics::Downforce::Downforce↔Point, [49](#)
- positionIntensity
 - NWH::VehiclePhysics::CameraOnboard, [35](#)
- positionSmoothing
 - NWH::VehiclePhysics::CameraOnboard, [35](#)
- postShiftBan
 - NWH::VehiclePhysics::Transmission, [149](#)
- Power

- NWH::VehiclePhysics::Engine, 59
- powerCoefficient
 - NWH::VehiclePhysics::Axle, 20
- powerCurve
 - NWH::VehiclePhysics::Engine, 57
- PowerGainMultiplier
 - NWH::VehiclePhysics::ForcedInduction, 70
- PowerInHP
 - NWH::VehiclePhysics::Engine, 59
- powerReduction
 - NWH::VehiclePhysics::TractionControl, 138
- PressEvent
 - NWH::VehiclePhysics::SteeringWheel, 130
- prevGroundPoint
 - NWH::WheelController3D::WheelController::↔
Wheel, 187
- prevLength
 - NWH::WheelController3D::WheelController::↔
Spring, 125
- prevOverflow
 - NWH::WheelController3D::WheelController::↔
Spring, 125
- prevPowerReduction
 - NWH::VehiclePhysics::TractionControl, 138
- prevVelocity
 - NWH::WheelController3D::WheelController::↔
Wheel, 187
- prevWorldPosition
 - NWH::WheelController3D::WheelController::↔
Wheel, 187
- previousCollisionHash
 - NWH::VehiclePhysics::DamageHandler, 44
- RPMPercent
 - NWH::VehiclePhysics::Engine, 59
- RPM
 - NWH::VehiclePhysics::Axle, 21
 - NWH::VehiclePhysics::Engine, 59
 - NWH::VehiclePhysics::Transmission, 152
 - NWH::VehiclePhysics::Wheel, 182
- Radius
 - NWH::VehiclePhysics::Wheel, 181
- radius
 - NWH::WheelController3D::WheelController, 200
- RandomClip
 - NWH::VehiclePhysics::SoundComponent, 122
- RawVertical
 - NWH::VehiclePhysics::InputStates, 90
- raycastHit
 - NWH::WheelController3D::WheelController::↔
WheelHit, 208
- rearLights
 - NWH::VehiclePhysics::Lights, 93
- ReconstructGearList
 - NWH::VehiclePhysics::Transmission, 146
- RegisterSources
 - NWH::VehiclePhysics::SoundComponent, 120
- ReleaseEvent
 - NWH::VehiclePhysics::SteeringWheel, 130
- renderers
 - NWH::VehiclePhysics::Backfire, 22
- Reset
 - NWH::VehiclePhysics::Metrics::Metric, 94
 - NWH::VehiclePhysics::VehicleController, 164
- ResetBrakes
 - NWH::VehiclePhysics::Wheel, 178
- ResetCOM
 - NWH::VehiclePhysics::VehicleController, 164
- residualAngularVelocity
 - NWH::WheelController3D::WheelController::↔
Wheel, 187
- returnToCenterSpeed
 - NWH::VehiclePhysics::SteeringWheel, 131
- ReverseGearCount
 - NWH::VehiclePhysics::Transmission, 152
- ReverseGears
 - NWH::VehiclePhysics::Transmission, 152
- ReverseRPM
 - NWH::VehiclePhysics::Transmission, 152
- ReverseTransmitRPM
 - NWH::VehiclePhysics::Transmission, 146
- ReverseTransmitTorque
 - NWH::VehiclePhysics::Transmission, 146
- rigging
 - NWH::VehiclePhysics::VehicleController, 167
- right
 - NWH::WheelController3D::WheelController::↔
Wheel, 187
- rightBlinker
 - NWH::VehiclePhysics::InputStates, 89
- rightBlinkers
 - NWH::VehiclePhysics::Lights, 93
- rightSprockets
 - NWH::VehiclePhysics::Tracks, 136
- rightTrackRenderer
 - NWH::VehiclePhysics::Tracks, 136
- rightWheel
 - NWH::VehiclePhysics::Axle, 20
- rim
 - NWH::WheelController3D::WheelController::↔
Wheel, 187
- rimCollider
 - NWH::WheelController3D::WheelController::↔
Wheel, 187
- rimOffset
 - NWH::WheelController3D::WheelController, 200
 - NWH::WheelController3D::WheelController::↔
Wheel, 187
- rotationAngle
 - NWH::WheelController3D::WheelController::↔
Wheel, 188
- rotationSpeed
 - NWH::VehiclePhysics::FlipOver, 68
- rpm
 - NWH::WheelController3D::WheelController, 201
 - NWH::WheelController3D::WheelController::↔
Wheel, 188

- RpmOverflow
 - NWH::VehiclePhysics::Engine, [59](#)
- runOnStartup
 - NWH::VehiclePhysics::Engine, [58](#)
- sFriction
 - NWH::WheelController3D::WheelController, [196](#)
- Sand
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [75](#)
- ScanIgnoreLayers
 - NWH::WheelController3D::WheelController, [201](#)
- SetActiveFrictionPreset
 - NWH::WheelController3D::WheelController, [194](#)
- SetAudioSourceDefaults
 - NWH::VehiclePhysics::Sound, [112](#)
- SetBrakeIntensity
 - NWH::VehiclePhysics::Wheel, [178](#)
- SetCamber
 - NWH::WheelController3D::WheelController, [194](#), [195](#)
- SetDefaults
 - NWH::VehiclePhysics::Sound, [112](#)
 - NWH::VehiclePhysics::VehicleController, [165](#)
- SetVolume
 - NWH::VehiclePhysics::SoundComponent, [120](#)
- ShiftDown
 - NWH::VehiclePhysics::InputStates, [90](#)
- shiftDuration
 - NWH::VehiclePhysics::Transmission, [149](#)
- shiftDurationRandomness
 - NWH::VehiclePhysics::Transmission, [149](#)
- ShiftInto
 - NWH::VehiclePhysics::Transmission, [146](#)
- shiftPointRandomness
 - NWH::VehiclePhysics::Transmission, [149](#)
- ShiftUp
 - NWH::VehiclePhysics::InputStates, [90](#)
- Shifting
 - NWH::VehiclePhysics::Transmission, [153](#)
- showCOM
 - NWH::VehiclePhysics::CenterOfMass, [36](#)
- Side
 - NWH::WheelController3D::WheelController, [192](#)
- sideFriction
 - NWH::WheelController3D::WheelController, [201](#)
- sideFrictionCurve
 - NWH::WheelController3D::WheelController↔
Inspector, [205](#)
- sideParams
 - NWH::WheelController3D::WheelController↔
Inspector, [205](#)
- sideSkidVolume
 - NWH::VehiclePhysics::SkidComponent, [104](#)
- SideSlip
 - NWH::VehiclePhysics::Wheel, [182](#)
- SideSlipPercent
 - NWH::VehiclePhysics::Wheel, [182](#)
- sideSlipThreshold
 - NWH::VehiclePhysics::VehicleController, [167](#)
- SideToSideScanResolution
 - NWH::WheelController3D::WheelController, [201](#)
- sidewaysDir
 - NWH::WheelController3D::WheelController::↔
WheelHit, [208](#)
- sidewaysSlip
 - NWH::WheelController3D::WheelController::↔
WheelHit, [208](#)
- singleRay
 - NWH::WheelController3D::WheelController, [197](#)
- singleWheelHit
 - NWH::WheelController3D::WheelController, [197](#)
- skidComponent
 - NWH::VehiclePhysics::Sound, [115](#)
- skidSoundComponent
 - NWH::VehiclePhysics::GroundDetection::Ground↔
Entity, [85](#)
- skidmarkMaterial
 - NWH::VehiclePhysics::GroundDetection::Ground↔
Entity, [85](#)
- skidmarkStrength
 - NWH::VehiclePhysics::Skidmarks, [109](#)
- skidmarks
 - NWH::VehiclePhysics::Effects, [54](#)
- slip
 - NWH::WheelController3D::WheelController::↔
Friction, [72](#)
- slipBasedSkidIntensity
 - NWH::VehiclePhysics::GroundDetection::Ground↔
Entity, [86](#)
- slipCoefficient
 - NWH::WheelController3D::WheelController::↔
Friction, [72](#)
- slipSensitiveSurfaceSound
 - NWH::VehiclePhysics::GroundDetection::Ground↔
Entity, [86](#)
- smoke
 - NWH::VehiclePhysics::SurfaceParticles, [133](#)
- smokeGo
 - NWH::VehiclePhysics::SurfaceParticles, [133](#)
- smokeIntensity
 - NWH::VehiclePhysics::GroundDetection::Ground↔
Entity, [86](#)
- smokePrefab
 - NWH::VehiclePhysics::GroundDetection, [83](#)
- SmoothForwardSlip
 - NWH::VehiclePhysics::Wheel, [182](#)
- SmoothRPM
 - NWH::VehiclePhysics::Axle, [21](#)
 - NWH::VehiclePhysics::Wheel, [182](#)
- SmoothSideSlip
 - NWH::VehiclePhysics::Wheel, [182](#)
- smoothing
 - NWH::VehiclePhysics::Brakes, [26](#)
 - NWH::VehiclePhysics::CameraFollow, [30](#)
 - NWH::VehiclePhysics::CameraMouseDrag, [33](#)

- NWH::VehiclePhysics::EngineIdleSoundComponent, [62](#)
- Snow
 - NWH::WheelController3D::WheelController::↔
FrictionPreset, [75](#)
- solidAxle
 - NWH::VehiclePhysics::Rigging, [102](#)
- soot
 - NWH::VehiclePhysics::ExhaustSmoke, [65](#)
- sootColor
 - NWH::VehiclePhysics::ExhaustSmoke, [66](#)
- sound
 - NWH::VehiclePhysics::VehicleController, [167](#)
- Source
 - NWH::VehiclePhysics::SoundComponent, [122](#)
- source
 - NWH::VehiclePhysics::GroundDetection::Sound↔
Component, [118](#)
- Sources
 - NWH::VehiclePhysics::SoundComponent, [123](#)
- sources
 - NWH::VehiclePhysics::SoundComponent, [121](#)
- spatialBlend
 - NWH::VehiclePhysics::Sound, [115](#)
- Speed
 - NWH::VehiclePhysics::VehicleController, [169](#)
- speed
 - NWH::WheelController3D::WheelController, [201](#)
 - NWH::WheelController3D::WheelController::↔
Friction, [72](#)
- SpeedKPH
 - NWH::VehiclePhysics::VehicleController, [169](#)
- speedLimiter
 - NWH::VehiclePhysics::VehicleController, [167](#)
- SpeedMPH
 - NWH::VehiclePhysics::VehicleController, [169](#)
- SpoolPercent
 - NWH::VehiclePhysics::ForcedInduction, [70](#)
- spoolUpTime
 - NWH::VehiclePhysics::ForcedInduction, [70](#)
- spring
 - NWH::WheelController3D::WheelController, [197](#)
- springBottomedOut
 - NWH::WheelController3D::WheelController, [201](#)
- springCompression
 - NWH::WheelController3D::WheelController, [202](#)
- springCurve
 - NWH::WheelController3D::WheelController, [202](#)
- springLength
 - NWH::WheelController3D::WheelController, [202](#)
- springMaximumForce
 - NWH::WheelController3D::WheelController, [202](#)
- springOverExtended
 - NWH::WheelController3D::WheelController, [202](#)
- SpringTravel
 - NWH::VehiclePhysics::Wheel, [183](#)
- springTravel
 - NWH::WheelController3D::WheelController, [202](#)
- springTravelPoint
 - NWH::WheelController3D::WheelController, [203](#)
- springVelocity
 - NWH::WheelController3D::WheelController, [203](#)
- stability
 - NWH::VehiclePhysics::DrivingAssists, [52](#)
- Start
 - NWH::VehiclePhysics::Engine, [56](#)
 - NWH::WheelController3D::WheelController, [195](#)
- startSize
 - NWH::VehiclePhysics::ExhaustSmoke, [66](#)
- starting
 - NWH::VehiclePhysics::Engine, [58](#)
- StartingPercent
 - NWH::VehiclePhysics::Engine, [60](#)
- Steer
 - NWH::VehiclePhysics::Steering, [127](#)
- SteerAngle
 - NWH::VehiclePhysics::Wheel, [183](#)
- steerAngle
 - NWH::WheelController3D::WheelController, [203](#)
 - NWH::WheelController3D::WheelController::↔
Wheel, [188](#)
- steerCoefficient
 - NWH::VehiclePhysics::Axle::Geometry, [81](#)
- steering
 - NWH::VehiclePhysics::VehicleController, [167](#)
- steeringWheel
 - NWH::VehiclePhysics::MobileInputManager, [100](#)
 - NWH::VehiclePhysics::Steering, [128](#)
- steeringWheelGraphic
 - NWH::VehiclePhysics::SteeringWheel, [131](#)
- steeringWheelTurnRatio
 - NWH::VehiclePhysics::Steering, [128](#)
- Stop
 - NWH::VehiclePhysics::Engine, [56](#)
- stopLights
 - NWH::VehiclePhysics::Lights, [93](#)
- stopping
 - NWH::VehiclePhysics::Engine, [58](#)
- StoppingPercent
 - NWH::VehiclePhysics::Engine, [60](#)
- SubArray
 - NWH::VehiclePhysics::SkidmarkGenerator, [106](#)
- surfaceComponent
 - NWH::VehiclePhysics::Sound, [115](#)
- surfaceNoiseMixerGroup
 - NWH::VehiclePhysics::Sound, [116](#)
- surfaceSoundComponent
 - NWH::VehiclePhysics::GroundDetection::Ground↔
Entity, [86](#)
- Suspend
 - NWH::VehiclePhysics::Wheel, [178](#)
- suspensionComponent
 - NWH::VehiclePhysics::Sound, [116](#)
- suspensionForce
 - NWH::WheelController3D::WheelController, [203](#)
- suspensionMixerGroup

- NWH::VehiclePhysics::Sound, [116](#)
- tags
 - NWH::VehiclePhysics::GroundDetection::Ground↔ Entity, [86](#)
- target
 - NWH::VehiclePhysics::CameraFollow, [30](#)
 - NWH::VehiclePhysics::CameraMouseDrag, [33](#)
- targetClutchRPM
 - NWH::VehiclePhysics::Transmission, [149](#)
- targetForwardOffset
 - NWH::VehiclePhysics::CameraFollow, [30](#)
- targetPoint
 - NWH::WheelController3D::WheelController::↔ Spring, [125](#)
- targetShiftDownRPM
 - NWH::VehiclePhysics::Transmission, [149](#)
- targetShiftUpRPM
 - NWH::VehiclePhysics::Transmission, [150](#)
- targetUpOffset
 - NWH::VehiclePhysics::CameraFollow, [31](#)
- TarmacDry
 - NWH::WheelController3D::WheelController::↔ FrictionPreset, [75](#)
- TarmacWet
 - NWH::WheelController3D::WheelController::↔ FrictionPreset, [76](#)
- tcs
 - NWH::VehiclePhysics::DrivingAssists, [52](#)
- TcsPowerReduction
 - NWH::VehiclePhysics::Engine, [60](#)
- terrainTextureIndices
 - NWH::VehiclePhysics::GroundDetection::Ground↔ Entity, [86](#)
- textureOffsetSpeedCoefficient
 - NWH::VehiclePhysics::Tracks, [136](#)
- ThrottleDown
 - NWH::VehiclePhysics::MobileInputManager, [99](#)
- throttleSmoothing
 - NWH::VehiclePhysics::Engine, [58](#)
- ThrottleUp
 - NWH::VehiclePhysics::MobileInputManager, [99](#)
- tiltSensitivity
 - NWH::VehiclePhysics::MobileInputManager, [100](#)
- timeout
 - NWH::VehiclePhysics::FlipOver, [68](#)
- tireLoad
 - NWH::WheelController3D::WheelController::↔ Wheel, [188](#)
- tireRadius
 - NWH::WheelController3D::WheelController, [203](#)
 - NWH::WheelController3D::WheelController::↔ Wheel, [188](#)
- tireWidth
 - NWH::WheelController3D::WheelController, [203](#)
- toeAngle
 - NWH::VehiclePhysics::Axle::Geometry, [81](#)
- Toggle
 - NWH::VehiclePhysics::Engine, [56](#)
- ToggleHighBeams
 - NWH::VehiclePhysics::MobileInputManager, [99](#)
- ToggleLowBeams
 - NWH::VehiclePhysics::MobileInputManager, [99](#)
- topSpeed
 - NWH::VehiclePhysics::Metrics, [96](#)
- Torque
 - NWH::VehiclePhysics::Engine, [60](#)
- TorqueSplit
 - NWH::VehiclePhysics::Axle, [18](#)
 - NWH::VehiclePhysics::Transmission, [146](#)
- totalDriftDistance
 - NWH::VehiclePhysics::Metrics, [97](#)
- totalDriftTime
 - NWH::VehiclePhysics::Metrics, [97](#)
- TotalPowerReduction
 - NWH::VehiclePhysics::Engine, [60](#)
- trackedSteerIntensity
 - NWH::VehiclePhysics::Steering, [129](#)
- trackedVehicle
 - NWH::VehiclePhysics::Tracks, [137](#)
 - NWH::WheelController3D::WheelController, [197](#)
- Tracks
 - NWH::WheelController3D::WheelController::↔ FrictionPreset, [76](#)
- tracks
 - NWH::VehiclePhysics::VehicleController, [167](#)
- trailer
 - NWH::VehiclePhysics::VehicleController, [167](#)
- TrailerAttachDetach
 - NWH::VehiclePhysics::MobileInputManager, [99](#)
- trailerAttachDetach
 - NWH::VehiclePhysics::InputStates, [90](#)
- trailerInRange
 - NWH::VehiclePhysics::TrailerHandler, [141](#)
- trailerStand
 - NWH::VehiclePhysics::TrailerHandler, [141](#)
- trailersTag
 - NWH::VehiclePhysics::TrailerHandler, [141](#)
- transmission
 - NWH::VehiclePhysics::VehicleController, [167](#)
- transmissionMixerGroup
 - NWH::VehiclePhysics::Sound, [116](#)
- TransmissionType
 - NWH::VehiclePhysics::Transmission, [144](#)
- transmissionType
 - NWH::VehiclePhysics::Transmission, [150](#)
- transmissionWhineComponent
 - NWH::VehiclePhysics::Sound, [116](#)
- TransmitRPM
 - NWH::VehiclePhysics::Transmission, [147](#)
- TransmitTorque
 - NWH::VehiclePhysics::Transmission, [147](#)
- turboFlutterComponent
 - NWH::VehiclePhysics::Sound, [116](#)
- turboMixerGroup
 - NWH::VehiclePhysics::Sound, [116](#)
- turboWhistleComponent

- NWH::VehiclePhysics::Sound, [117](#)
- TurnOff
 - NWH::VehiclePhysics::Lights::VehicleLight, [173](#)
- TurnOffAllLights
 - NWH::VehiclePhysics::Lights, [92](#)
- TurnOn
 - NWH::VehiclePhysics::Lights::VehicleLight, [174](#)
- unitBumpForce
 - NWH::WheelController3D::WheelController::↔
Damper, [46](#)
- unitReboundForce
 - NWH::WheelController3D::WheelController::↔
Damper, [46](#)
- up
 - NWH::WheelController3D::WheelController::↔
Wheel, [188](#)
- Update
 - NWH::VehiclePhysics::ABS, [15](#)
 - NWH::VehiclePhysics::AirBrakeComponent, [16](#)
 - NWH::VehiclePhysics::Axle, [19](#)
 - NWH::VehiclePhysics::Backfire, [22](#)
 - NWH::VehiclePhysics::BackfireComponent, [23](#)
 - NWH::VehiclePhysics::BlinkerComponent, [24](#)
 - NWH::VehiclePhysics::Brakes, [25](#)
 - NWH::VehiclePhysics::CrashComponent, [40](#)
 - NWH::VehiclePhysics::DamageHandler, [42](#)
 - NWH::VehiclePhysics::DrivingAssists, [52](#)
 - NWH::VehiclePhysics::Effects, [53](#)
 - NWH::VehiclePhysics::Engine, [56](#)
 - NWH::VehiclePhysics::EngineIdleSoundComponent, [61](#)
 - NWH::VehiclePhysics::EngineStartStopComponent, [63](#)
 - NWH::VehiclePhysics::ExhaustSmoke, [64](#)
 - NWH::VehiclePhysics::FlipOver, [67](#)
 - NWH::VehiclePhysics::ForcedInduction, [69](#)
 - NWH::VehiclePhysics::Fuel, [77](#)
 - NWH::VehiclePhysics::GearChangeComponent, [80](#)
 - NWH::VehiclePhysics::HornComponent, [87](#)
 - NWH::VehiclePhysics::Lights, [92](#)
 - NWH::VehiclePhysics::Metrics, [96](#)
 - NWH::VehiclePhysics::Metrics::Metric, [94](#)
 - NWH::VehiclePhysics::Rigging, [102](#)
 - NWH::VehiclePhysics::SkidComponent, [103](#)
 - NWH::VehiclePhysics::SkidmarkGenerator, [106](#)
 - NWH::VehiclePhysics::Skidmarks, [108](#)
 - NWH::VehiclePhysics::Sound, [112](#)
 - NWH::VehiclePhysics::SoundComponent, [121](#)
 - NWH::VehiclePhysics::Stability, [126](#)
 - NWH::VehiclePhysics::SurfaceComponent, [132](#)
 - NWH::VehiclePhysics::SurfaceParticles, [133](#)
 - NWH::VehiclePhysics::SuspensionComponent, [134](#)
 - NWH::VehiclePhysics::Tracks, [136](#)
 - NWH::VehiclePhysics::TractionControl, [138](#)
 - NWH::VehiclePhysics::TrailerHandler, [139](#)
 - NWH::VehiclePhysics::Transmission, [147](#)
 - NWH::VehiclePhysics::TransmissionWhine↔
Component, [154](#)
 - NWH::VehiclePhysics::TurboFlutterComponent, [155](#)
 - NWH::VehiclePhysics::TurboWhistleComponent, [156](#)
 - NWH::VehiclePhysics::VehicleController, [165](#)
 - NWH::VehiclePhysics::Wheel, [178](#)
 - UpdateClutch
 - NWH::VehiclePhysics::Transmission, [147](#)
 - UpdateDelegate
 - NWH::VehiclePhysics::Metrics::Metric, [94](#)
 - UpdateSteering
 - NWH::VehiclePhysics::MobileInputManager, [100](#)
 - useForcedInduction
 - NWH::VehiclePhysics::ForcedInduction, [70](#)
 - useFuel
 - NWH::VehiclePhysics::Fuel, [78](#)
 - useRimCollider
 - NWH::WheelController3D::WheelController, [197](#)
 - valid
 - NWH::WheelController3D::WheelController::↔
WheelHit, [208](#)
 - value
 - NWH::VehiclePhysics::Metrics::Metric, [95](#)
 - vaporColor
 - NWH::VehiclePhysics::ExhaustSmoke, [66](#)
 - vc
 - NWH::VehiclePhysics::SoundComponent, [122](#)
 - vehicleCameras
 - NWH::VehiclePhysics::CameraChanger, [28](#)
 - vehicleChanger
 - NWH::VehiclePhysics::DesktopInputManager, [47](#)
 - NWH::VehiclePhysics::MobileInputManager, [100](#)
 - VehicleCollisionState
 - NWH::VehiclePhysics::VehicleController, [162](#)
 - vehicleController
 - NWH::VehiclePhysics::CameraOnboard, [35](#)
 - NWH::VehiclePhysics::DesktopInputManager, [47](#)
 - NWH::VehiclePhysics::MobileInputManager, [100](#)
 - vehicleMass
 - NWH::VehiclePhysics::VehicleFromScript, [172](#)
 - vehicleRigidbody
 - NWH::VehiclePhysics::VehicleController, [168](#)
 - VehicleSide
 - NWH::WheelController3D::WheelController, [204](#)
 - vehicleTag
 - NWH::VehiclePhysics::VehicleChanger, [158](#)
 - vehicles
 - NWH::VehiclePhysics::VehicleChanger, [158](#)
 - velocity
 - NWH::WheelController3D::WheelController::↔
Spring, [125](#)
 - NWH::WheelController3D::WheelController::↔
Wheel, [188](#)
 - Vertical
 - NWH::VehiclePhysics::InputStates, [91](#)
 - verticalMaxAngle

- NWH::VehiclePhysics::CameraMouseDrag, [34](#)
- verticalMinAngle
 - NWH::VehiclePhysics::CameraMouseDrag, [34](#)
- verticalMouseSensitivity
 - NWH::VehiclePhysics::CameraMouseDrag, [34](#)
- Visual
 - NWH::WheelController3D::WheelController, [204](#)
- visual
 - NWH::WheelController3D::WheelController::↔
Wheel, [188](#)
- VisualTransform
 - NWH::VehiclePhysics::Wheel, [183](#)
- volume
 - NWH::VehiclePhysics::GroundDetection::Sound↔
Component, [118](#)
 - NWH::VehiclePhysics::SoundComponent, [122](#)
- volumeRange
 - NWH::VehiclePhysics::EngineIdleSoundComponent,
[62](#)
- weight
 - NWH::WheelController3D::WheelController::↔
WheelHit, [208](#)
- Wheel
 - NWH::VehiclePhysics::Wheel, [177](#)
- wheel
 - NWH::VehiclePhysics::SkidmarkGenerator, [107](#)
 - NWH::WheelController3D::WheelController, [197](#)
- wheelBones
 - NWH::VehiclePhysics::Rigging, [102](#)
- WheelController
 - NWH::VehiclePhysics::Wheel, [183](#)
- wheelController
 - NWH::VehiclePhysics::Wheel, [179](#)
- WheelHit
 - NWH::WheelController3D::WheelController::↔
WheelHit, [206](#)
- wheelHit
 - NWH::WheelController3D::WheelController, [197](#)
- wheelRadius
 - NWH::VehiclePhysics::VehicleFromScript, [172](#)
- WheelSkid
 - NWH::VehiclePhysics::VehicleController, [170](#)
- WheelSpeed
 - NWH::VehiclePhysics::VehicleController, [170](#)
- WheelSpin
 - NWH::VehiclePhysics::Axle, [21](#)
 - NWH::VehiclePhysics::VehicleController, [170](#)
- wheelTag
 - NWH::VehiclePhysics::VehicleFromScript, [172](#)
- wheelWidth
 - NWH::VehiclePhysics::VehicleFromScript, [172](#)
- Wheels
 - NWH::VehiclePhysics::VehicleController, [170](#)
- Width
 - NWH::VehiclePhysics::Wheel, [183](#)
- width
 - NWH::WheelController3D::WheelController::↔
Wheel, [189](#)
- worldPosition
 - NWH::WheelController3D::WheelController::↔
Wheel, [189](#)
- worldRotation
 - NWH::WheelController3D::WheelController::↔
Wheel, [189](#)