

ModelDesk

Scenario Creation

For ModelDesk 5.5

Release 2021-A – May 2021

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: <http://www.dspace.com/go/locations>
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: <http://www.dspace.com/go/supportrequest>. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit <http://www.dspace.com/go/patches> for software updates and patches.

Important Notice

This publication contains proprietary information that is protected by copyright. All rights are reserved. The publication may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the publication must not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© 2018 - 2021 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

This publication and the contents hereof are subject to change without notice.

AUTERA, ConfigurationDesk, ControlDesk, MicroAutoBox, MicroLabBox, SCALEXIO, SIMPHERA, SYNECT, SystemDesk, TargetLink and VEOS are registered trademarks of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

Contents

About This Document	17
Basics and Instructions	19
Introduction to the Scenario Editor	20
Basics of the Scenario Editor	20
Graphical User Interface of the Scenario Editor	22
Filtering, Collapsing, or Expanding Blocks in the Scenario Editor	24
Basics of Scenarios	26
Scenarios	26
Basics of Maneuvers	27
Basics of Fellows	29
Basics of Global User Signals	31
Basics of Activities	31
Basics of Transitions	36
Example of Scenarios with Roads	38
Creating Maneuvers	39
Workflow for Maneuver Creation	40
How to Start Scenario Creation	41
How to Add Sequences and Segments for the Maneuver	43
How to Accelerate and Brake in a Maneuver	45
How to Shift Gears in a Maneuver	46
How to Follow a Road Automatically	48
How to Specify a Lane Change Maneuver	50
How to Steer in a Maneuver	52
How to Specify a Transition for a Maneuver	53
How to Specify the Properties of the Driver Model	54
How to Select and Specify Maneuver User Signals	56
Creating Traffic of Scenarios	59
Workflow for Traffic Creation	59
How to Create a Fellow User Signal	60
How to Create and Specify a Fellow	62
How to Initialize and Specify the Movements of a Fellow	63
How to Specify a Global User Signal	67
How to Specify the Properties of a Traffic Driver	69

Working with Scenarios.....	71
Using Scenarios.....	71
Modifying Properties of a Scenario via Alias Variables.....	72
Controlling Maneuvers and Driving Cycles.....	74
Basics of Controlling Maneuvers and Driving Cycles.....	74
How to Control a Maneuver or a Driving Cycle.....	76

Reference Information 79

Scenario Editor Management Commands.....	80
Activate.....	81
Activate from Pool.....	81
Download / Download Scenario.....	82
Download Maneuver.....	83
Download Traffic.....	83
Generate Traffic Ini.....	84
New.....	85
New (Traffic Driver).....	86
Open.....	86
Open (Traffic Driver).....	87
Open from Pool.....	87
Remove.....	88
Save.....	89
Save As.....	89
Scenario Editor Dialogs and Panes.....	91
Edit Table Dialog.....	91
Environment Configuration Dialog.....	92
Scenario Configuration Page.....	94
Scenario Pane.....	96
Scenario Properties Dialog.....	98
Scenario Settings Dialog.....	100
Traffic Driver Dialog.....	101
Scenario Editor Creation Commands.....	103
Append Segment.....	104
Append Sequence.....	105
Collapse.....	105
Collapse All.....	106
Copy.....	107
Cut.....	107
Delete <Element>.....	108
Delete Sequence.....	108

Expand.....	109
Expand All.....	110
Insert Segment - After.....	110
Insert Segment - Before.....	111
Layout Diagram.....	111
New – Fellow.....	112
New – Maneuver.....	113
New – User Signal.....	114
Paste.....	115
Paste Sequence.....	115
Properties.....	116
Settings.....	116
Scenario Editor Properties.....	118
Activity Properties (Fellow).....	119
Activity Properties (Global User Signal).....	120
Activity Properties (Maneuver).....	121
Clutch and Gear (MT) Properties.....	122
Fellow Properties.....	126
Fellow User Signal Properties.....	127
Global User Signal Properties.....	129
Lateral Type Properties (Fellow).....	129
Lateral Type Properties (Maneuver).....	133
Longitudinal Type Properties (Fellow).....	143
Longitudinal Type Properties (Maneuver).....	151
Maneuver Properties.....	158
Maneuver User Signal Properties.....	159
Segment Properties.....	160
Selector Lever (AT) Properties.....	161
Start Properties (Fellow).....	163
Start Properties (Global User Signal).....	164
Start Properties (Maneuver).....	164
Traffic Scenario Properties.....	166
Transition Properties.....	167
Source Properties.....	172
Maneuver Control Commands.....	175
Reset.....	175
Start.....	176
Stop.....	177

Automation	179
Programming ModelDesk Automation.....	180
Automating Scenarios in Python.....	180
Automated Maneuver Control in Python.....	184
Overview of the Classes for Creating Scenarios.....	187
Overview of the Object Model for Model Configuration.....	191
Overview of the Object Model for Configuring Traffic Drivers.....	192
Classes for Maneuver Control.....	193
ManeuverControl.....	193
Class Description (ManeuverControl).....	194
Start.....	195
Reset.....	196
Stop.....	196
Classes for Creating Scenarios.....	198
ActiveTrafficScenario.....	198
Class Description (ActiveTrafficScenario).....	199
CheckConsistency.....	200
Download.....	201
DownloadContent.....	202
GenerateModelIni.....	202
RemoveRoad.....	203
RemoveRoad.....	204
ReplaceContent.....	204
Save.....	205
SaveAs.....	205
DriverProperties.....	206
Class Description (DriverProperties).....	206
ManeuverProperties.....	207
Class Description (ManeuverProperties).....	207
ScenarioProperties.....	208
Class Description (ScenarioProperties).....	208
Reset.....	209
TrafficProperties.....	210
Class Description (TrafficProperties).....	210

Classes for Configuring the Maneuver.....	211
Driver.....	212
Class Description (Driver).....	212
DriverType.....	213
Class Description (DriverType).....	213
GearProfile.....	214
Class Description (GearProfile).....	214
Activate.....	215
LateralProfile.....	216
Class Description (LateralProfile).....	216
Activate.....	217
LongitudinalProfile.....	218
Class Description (LongitudinalProfile (Maneuver)).....	218
Activate.....	219
Maneuver.....	219
Class Description (Maneuver).....	219
Maneuvers.....	220
Class Description (Maneuvers).....	221
Add.....	221
Item.....	222
Remove.....	223
ManeuverActivity.....	223
Class Description (ManeuverActivity).....	224
ManeuverSegment.....	224
Class Description (ManeuverSegment).....	224
ManeuverSegments.....	225
Class Description (ManeuverSegments).....	226
Append.....	227
Insert.....	227
Item.....	228
Remove.....	229
ManeuverSequence.....	229
Class Description (ManeuverSequence).....	230
ManeuverSequences.....	231
Class Description (ManeuverSequences).....	231
Add.....	232

Item.....	232
Remove.....	233
ManeuverTransition.....	234
Class Description (ManeuverTransition).....	234
OpenClutch.....	235
Class Description (OpenClutch).....	235
ReferenceGear.....	235
Class Description (ReferenceGear).....	235
SelectorProfile.....	236
Class Description (SelectorProfile).....	237
Activate.....	237
Stimulus.....	238
Class Description (Stimulus).....	238
Classes for Configuring Fellows.....	240
Fellow.....	241
Class Description (Fellow).....	241
FellowActivity.....	242
Class Description (FellowActivity).....	242
Fellows.....	242
Class Description (Fellows).....	243
Add.....	243
Item.....	244
Remove.....	245
FellowSegment.....	246
Class Description (FellowSegment).....	246
FellowSegments.....	246
Class Description (FellowSegments).....	247
Append.....	248
Insert.....	248
Item.....	249
Remove.....	250
FellowSequence.....	251
Class Description (FellowSequence).....	251
FellowSequences.....	251
Class Description (FellowSequences (Collection)).....	252
Add.....	253

Item.....	253
Remove.....	254
FellowTransition.....	255
Class Description (FellowTransition).....	255
LateralProfile.....	256
Class Description (LateralProfile).....	256
Activate.....	257
LongitudinalProfile.....	258
Class Description (LongitudinalProfile).....	258
Activate.....	259
TrafficObjectTypes.....	260
Class Description (TrafficObjectTypes).....	260
Activate.....	261
TrafficObjectType.....	261
Class Description (TrafficObjectType).....	262
UserOutputs.....	262
Class Description (UserOutputs (Collection)).....	262
Item.....	263
UserOutput.....	264
Class Description (UserOutput).....	264
Classes for Configuring Global User Signals.....	265
GlobalUserSignals.....	266
Class Description (GlobalUserSignals).....	266
Add.....	267
Item.....	267
Remove.....	268
GlobalUserSignal.....	269
Class Description (GlobalUserSignal).....	269
UserSignalSegments.....	270
Class Description (UserSignalSegments).....	270
Append.....	271
Insert.....	272
Item.....	272
Remove.....	273
UserSignalSegment.....	274
Class Description (UserSignalSegment).....	274

UserSignalSequences.....	275
Class Description (UserSignalSequences).....	275
Add.....	276
Item.....	277
Remove.....	277
UserSignalSequence.....	278
Class Description (UserSignalSequence).....	278
UserSignalTransition.....	279
Class Description (UserSignalTransition).....	279
UserSignalValue.....	280
Class Description (UserSignalValue).....	280
Activate.....	281
Classes for Configuring Conditions.....	282
ConditionDistance.....	283
Class Description (ConditionDistance).....	283
ConditionDistanceRefLine.....	283
Class Description (ConditionDistanceRefLine).....	284
ConditionDrivenDistance.....	284
Class Description (ConditionDrivenDistance).....	285
ConditionDuration.....	285
Class Description (ConditionDuration).....	285
ConditionEndless.....	286
Class Description (ConditionEndless).....	286
ConditionImmediate.....	287
Class Description (ConditionImmediate).....	287
ConditionLateralAcceleration.....	287
Class Description (ConditionLateralAcceleration).....	287
ConditionPosition.....	288
Class Description (ConditionPosition).....	288
ConditionTimegap.....	289
Class Description (ConditionTimegap).....	289
ConditionTrigger.....	290
Class Description (ConditionTrigger).....	290
ConditionVelocity.....	290
Class Description (ConditionVelocity).....	291

Continue.....	291
Class Description (Continue (LateralProfile)).....	291
Continue.....	292
Class Description (Continue (LongitudinalProfile)).....	292
TransitionsConditions.....	293
Class Description (TransitionsConditions).....	293
Add.....	294
FindItem.....	295
Item.....	296
Remove.....	297
Classes for Model Configuration.....	298
ListElements.....	298
Class Description (ListElements).....	299
Add.....	300
Item.....	300
Move.....	301
Remove.....	302
ListElement.....	302
Class Description (ListElement).....	302
ManeuverConfiguration.....	303
Class Description (ManeuverConfiguration).....	303
ModelConfiguration.....	304
Class Description (ModelConfiguration).....	304
Reset.....	305
ReadOnlyListElements.....	306
Class Description (ReadOnlyListElements).....	306
Item.....	307
Move.....	308
ScenarioConfiguration.....	308
Class Description (ScenarioConfiguration).....	308
Classes for Configuring Traffic Drivers.....	310
DrivingStyle.....	310
Class Description (DrivingStyle).....	310
RuleCompliantDriving.....	311
Class Description (RuleCompliantDriving).....	311

TrafficDrivers.....	313
Class Description (TrafficDrivers).....	313
Item.....	314
TrafficDriverTypes.....	314
Class Description (TrafficDriverTypes).....	315
Activate.....	315
TrafficDriver.....	316
Class Description (TrafficDriver).....	316
DiscardChanges.....	317
Save.....	318
SaveAs.....	319
TrafficSignRecognition.....	319
Class Description (TrafficSignRecognition).....	319
Miscellaneous Classes for Scenarios.....	321
Acceleration.....	323
Class Description (Acceleration).....	323
AccelerationSource.....	324
Class Description (AccelerationSource).....	324
Activate.....	325
ApproximationOperators.....	326
Class Description (ApproximationOperators).....	326
Activate.....	327
ApproximationOperator.....	327
Class Description (ApproximationOperator).....	328
BasicRoad.....	328
Class Description (BasicRoad).....	328
DependencyTypes.....	329
Class Description (DependencyTypes).....	329
Activate.....	330
DependencyTypeAbsolute.....	330
Class Description (DependencyTypeAbsolute).....	331
DependencyTypeRelative.....	331
Class Description (DependencyTypeRelative).....	331
DistanceMeter.....	332
Class Description (DistanceMeter).....	332

DistanceMeterSource.....	333
Class Description (DistanceMeterSource).....	333
Activate.....	334
DistanceTime.....	334
Class Description (DistanceTime).....	335
DistanceTimeSource.....	335
Class Description (DistanceTimeSource).....	336
Activate.....	336
FollowRoad.....	337
Class Description (FollowRoad).....	337
LateralAcceleration.....	338
Class Description (LateralAcceleration).....	338
LateralDeviation.....	339
Class Description (LateralDeviation).....	339
LateralDeviationSource.....	339
Class Description (LateralDeviationSource).....	340
Activate.....	341
LaneSelection.....	341
Class Description (LaneSelection).....	342
LaneSelectionSource.....	342
Class Description (LaneSelectionSource).....	343
Activate.....	343
NamedItem.....	344
Class Description (NamedItem).....	344
NamedItemList.....	345
Class Description (NamedItemList).....	345
Add.....	346
Item.....	346
Remove.....	347
NotUsed.....	348
Class Description (NotUsed (LateralProfile)).....	348
NotUsed.....	348
Class Description (NotUsed (LongitudinalProfile)).....	349
PedalStimulus.....	349
Class Description (PedalStimulus).....	349

Position.....	350
Class Description (Position).....	350
PositionSource.....	351
Class Description (PositionSource).....	351
Activate.....	352
RelativeOperators.....	352
Class Description (RelativeOperator).....	352
RelativeOperators.....	353
Class Description (RelativeOperators).....	353
Activate.....	354
RelativeObject.....	355
Class Description (RelativeObject).....	355
RelativeVehicleSelection.....	355
Class Description (RelativeVehicleSelection).....	356
Activate.....	356
Route.....	357
Class Description (Route).....	357
RouteSelection.....	358
Class Description (RouteSelection).....	358
Activate.....	359
SourceTypes.....	360
Circle.....	361
Constant.....	362
Extern.....	362
FinalDeviation.....	362
FinalDistance.....	363
FinalLane.....	363
FinalVelocity.....	364
Fixed.....	364
LaneChange.....	365
Manual.....	365
Pulse.....	365
Ramp.....	366
SimpleConstant.....	366
Sine.....	367
SineWithDwell.....	367
SmoothLaneChange.....	368
Step.....	368
StopWithin.....	369

Straight.....	369
Table.....	370
UntilStop.....	370
YawAngleConstant.....	370
Standstill.....	371
Class Description (Standstill).....	371
StimulusAngle.....	371
Class Description (StimulusAngle).....	372
StimulusTorque.....	372
Class Description (StimulusTorque).....	372
Stop.....	373
Class Description (Stop).....	373
StopWithin.....	374
Class Description (StopWithin).....	374
StopWithinSource.....	374
Class Description (StopWithinSource).....	375
Activate.....	375
Table.....	376
Class Description (Table).....	377
Export.....	377
Import.....	378
Insert.....	379
Item.....	380
Remove.....	380
TableValue.....	381
Class Description (TableValue).....	381
Velocity.....	382
Class Description (Velocity).....	382
VelocitySource.....	383
Class Description (VelocitySource).....	383
Activate.....	384
Constants for Scenarios.....	385
Constants for Scenarios.....	385
Limitations	391
Limitations of the Scenario Editor.....	391

Index

393









About This Document

Contents

This document introduces you to the Scenario Editor that is used to create maneuvers and traffic scenarios.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
 DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
 WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
 CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
 NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
 Note	Indicates important information that you should take into account to avoid malfunctions.
 Tip	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>

or

%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>

Documents folder A standard folder for user-specific documents.

%USERPROFILE%\Documents\dSPACE\<ProductName>\

<VersionNumber>

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\

<ProductName>

Accessing dSPACE Help and PDF Files


After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the  icon in dSPACE Help. The PDF opens on the first page.

Basics and Instructions

Where to go from here

Information in this section

Introduction to the Scenario Editor.....	20
The introduction gives you basic information on the Scenario Editor.	
Basics of Scenarios.....	26
Provides basic information on the specification of a scenario.	
Creating Maneuvers.....	39
A maneuver specifies the movement of the ASM vehicle.	
Creating Traffic of Scenarios.....	59
Shows how to create a traffic of a scenario by specifying all the fellows and global user signals.	
Working with Scenarios.....	71
You can connect the scenarios you created to roads and work with them in ModelDesk.	
Controlling Maneuvers and Driving Cycles.....	74
You can use ModelDesk to start, stop, or reset maneuvers for vehicle dynamics test or driving cycles for engine tests.	

Introduction to the Scenario Editor

Introduction The introduction gives you basic information on the Scenario Editor.

Where to go from here

Information in this section

[Basics of the Scenario Editor..... 20](#)

With the Scenario Editor, you can create maneuvers and traffic scenarios for controlling the movement of the ASM vehicle and simulating traffic situations such as vehicles overtaking and changing lanes.

[Graphical User Interface of the Scenario Editor..... 22](#)

Describes all the elements of the Scenario Editor.

[Filtering, Collapsing, or Expanding Blocks in the Scenario Editor..... 24](#)

To focus on the blocks you are working on, you can reduce the number of visible blocks and collapse other blocks.

Basics of the Scenario Editor

Introduction With the Scenario Editor, you can create maneuvers and traffic scenarios for controlling the movement of the ASM vehicle and simulating traffic situations such as vehicles overtaking and changing lanes.

Features

When you create a scenario with the Scenario Editor, you can:

- Specify the movement of the ASM vehicle.
- Specify the movements of the traffic participants, such as fellow vehicles or pedestrians
- Specify the road the scenario is linked to
- Specify the transition from one activity to the next
- Specify user signals for the maneuver, fellows, or globally to communicate with the simulation model
- Download the scenario parameters to Simulink or generate a parameter file to be loaded to the simulation platform via an experiment software

Definitions

Scenario A scenario contains a maneuver and an optional traffic scenario. Depending on your test scenario, you can define only the movement of the ASM

vehicle (maneuver) or additionally the movement of traffic objects around the ASM vehicle (traffic scenario).

Maneuver A maneuver defines the movement of the ASM vehicle. A maneuver can consist of several sequences that are executed one after the other. You can specify specific start properties, for each sequence. Each sequence has a start segment to specify the maneuver start. After a start segment, several segments can be specified. Each segment consists of an activity and a transition. In an activity, one kind of movement is specified. A transition contains one or more conditions that stops the activity.

Traffic A traffic defines the movements of objects (fellows) around the ASM vehicle. Traffics consist of several segments.

ASM vehicle The ASM vehicle is the initial vehicle in a scenario. Its movement is specified by the maneuver.

Fellow A fellow vehicle (fellow) is a vehicle defined in the traffic. Its movement is defined by specifying one or more sequences for it. To visualize the fellow, a suitable traffic object can be selected.

User signal User signals can be used to communicate with the maneuver scheduler in the simulation model by triggering user-defined actions. You can specify user signals globally in the scenario (global user signals) or as part of the maneuver or a fellow definition (maneuver user signal or fellow user signal). Global user signals are specified in the same way as fellow definitions.

Activity An activity describes the behavior of the maneuver, a fellow, or global user signal. The behavior of the maneuver and a fellow can be described in longitudinal and lateral direction. This means that you can specify the acceleration of a fellow or a lane change. For global user signals, only a value can be specified.

The activity can be defined with absolute values or relative to the ASM vehicle or other traffic participants.


Transition A transition contains one or more conditions from that one must be met to end the current activity of the maneuver, a fellow, or global user signal and start the next one.

Segment A segment is the combination of an activity and a transition. You should use the first segment for initializing the maneuver, fellow, or global user signal.

Sequence A sequence is a chain of segments. You can specify several segments, each containing an activity and a transition. The segments are connected in a chain so that the ASM vehicle and fellows can make different kinds of movements or the global user signals can have different values in the scenario. The first segment of a sequence is used for initializing the maneuver or fellow.

Start segment In the start segment of the maneuver or a fellow sequence you can specify the route and direction of travel.

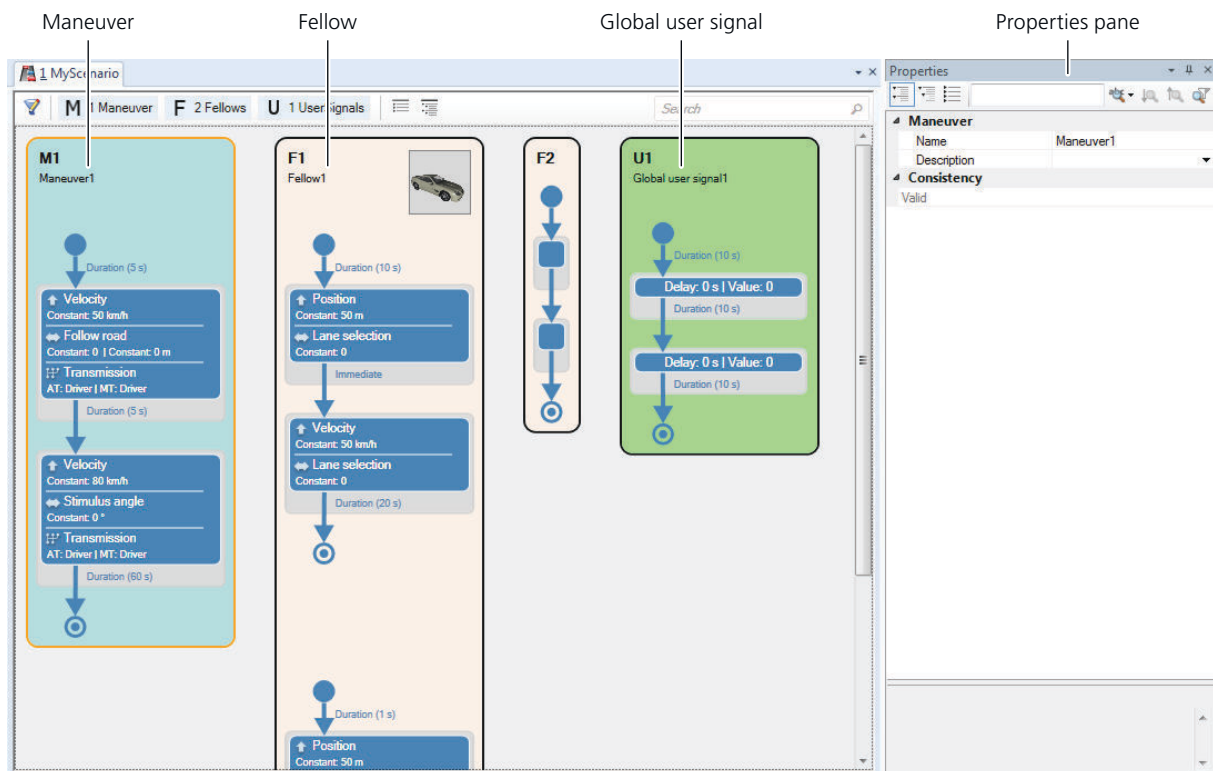
Example

For an example of traffic scenario and its modeling, refer to [Example of a Scenario](#) (ASM Traffic Reference )

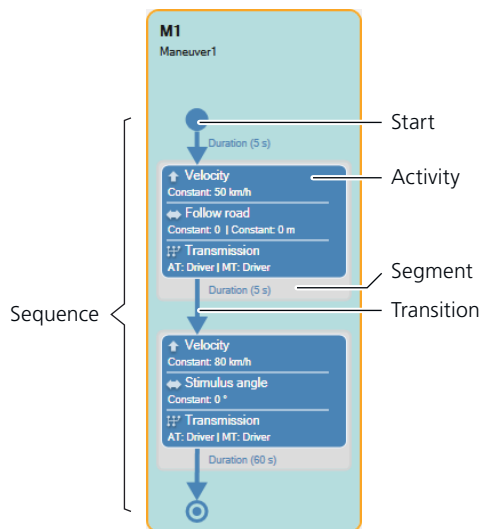
Graphical User Interface of the Scenario Editor

Overview of the graphical user interface

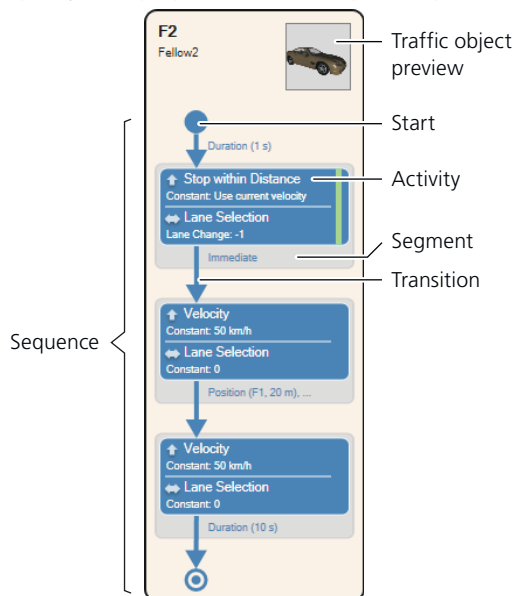
The Scenario Editor is integrated in ModelDesk's main window.



Maneuver block The Maneuver block is the graphical representation of sequences of the maneuver of the ASM vehicle. You can click the contained elements, such as activity or transition, and specify their properties in the Properties pane.



Fellow block The fellow block is the graphical representation of sequences of a fellow. You can click the contained elements, such as activity or transition, and specify their properties in the Properties pane.



Global user signal block The global user signal block is the graphical representation of a global user signal. You can click the contained elements, such as activity or transition, and specify their properties in the Properties pane. You can also click an element's name to modify it.

Context menu The Scenario Editor has a context menu on the workspace. The context menu contains various commands for editing the maneuver, fellow definition, and global user signal. The commands that are available depend on the selected element.

Properties pane Displays and lets you specify properties for different elements. The properties are grouped in categories. The contents of the pane

depend on the pane that is currently active. The **Properties** pane can be hidden or shown by using the **Switch Controlbars** command in the **View - Controlbar** ribbon group.

Related topics

Basics

[Filtering, Collapsing, or Expanding Blocks in the Scenario Editor..... 24](#)

References

[Layout Diagram..... 111](#)

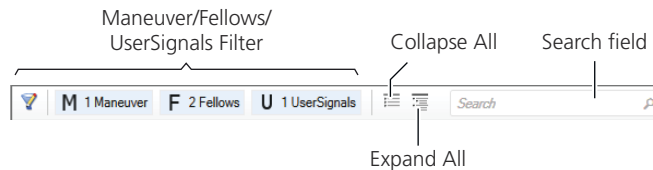
Filtering, Collapsing, or Expanding Blocks in the Scenario Editor

Introduction

To focus on the blocks you are working on, you can reduce the number of visible blocks and collapse other blocks.

Filter and collapse/expand

The Scenario Editor has some commands that you can use to filter or collapse/expand the maneuver, fellow and global user signal blocks.



Maneuver/Fellows/UserSignals Filter Use the Maneuver, Fellows, and Global User Signals Filter to hide or display all the blocks of this type. When a filter is active, the Scenario Editor displays the number of visible blocks and the total number of blocks.

Collapse All/Expand All buttons Use the Collapse All button to collapse all the blocks.

Use the Expand All button to expand all the blocks.

Tip

You can also double-click a block to collapse/expand this block individually.

Search field Use the search field to filter all the Maneuver, Fellow, and Global User Signal blocks. When you enter a string, the Scenario Editor displays only the blocks whose names or values contain the search string.

Related topics

References

Collapse.....	105
Collapse All.....	106
Expand.....	109
Expand All.....	110

Basics of Scenarios

Introduction Scenarios define the movement of the ASM vehicle and the movements of fellows.

Where to go from here

Information in this section

Scenarios.....	26
Scenarios define the movement of the ASM vehicle and the movements of fellows relative to the ASM vehicle or to the road.	
Basics of Maneuvers.....	27
A maneuver specifies the movement of the ASM vehicle in a scenario.	
Basics of Fellows.....	29
A fellow is a vehicle or a pedestrian whose movements are defined in a scenario.	
Basics of Global User Signals.....	31
You can specify global user signals independently of the maneuver and fellow definitions.	
Basics of Activities.....	31
An activity specifies the movement of the maneuver or a fellow or the value of a global user signal.	
Basics of Transitions.....	36
Transitions specify the conditions that end one activity and start the following activity.	
Example of Scenarios with Roads.....	38
A scenario can be combined with a road. Only if it combined with the correct road, the simulation results make sense.	

Scenarios

Introduction Scenarios define the movement of the ASM vehicle and the movements of fellows relative to the ASM vehicle or to the road. You can simulate traffic situations such as vehicles overtaking and changing lanes.

Connection to road A scenario can be connected to a road network. So it can use all the specification of the road, for example, the route that can be driven by the ASM vehicle and the fellows or the lanes that can be selected.

If you want to specify a maneuver only, i.e., the stimulation and movements of the ASM vehicle, connecting to a road network is optional. If you want to specify the movements of fellows, connecting to a road network is mandatory. In addition, you must assign routes to the fellows. Therefore, routes must be specified for the road network.

Use the Road Generator to specify a road network for the scenario, refer to [Introduction to the Road Generator \(ModelDesk Road Creation !\[\]\(c507f772dba2b921f86777f01218e570_img.jpg\)](#)).

It is also possible to specify a maneuver without connection to a road.

Traffic scenario components

A scenario consists of a maneuver, fellows and global user signals:

- A maneuver defines the movement of the ASM vehicle. You can specify only one maneuver in a scenario. For details, refer to [Basics of Maneuvers](#) on page 27.
- A fellow defines the movement of a traffic participant. Fellows are optional. You can specify any number of fellows in a scenario. For details, refer to [Basics of Fellows](#) on page 29.
- A global user signal defines a signal that can be read in the simulation model and is independent of the maneuver and fellow definition. Global user signals are optional. You can specify any number of global user signals in a scenario. For details, refer to [Basics of Global User Signals](#) on page 31.

A segment describes the maneuver, a fellow or the value of a global user signal. A segment has an activity and a transition. The activity specifies the kind of movement or value. The transition specifies when the current activity ends and the next activity starts. Refer to:

- [Basics of Activities](#) on page 31
- [Basics of Transitions](#) on page 36

Related topics

Basics

[Basics of the Scenario Editor..... 20](#)


References

[Environment Configuration Dialog..... 92](#)

Basics of Maneuvers

Introduction

In scenarios, you can simulate the movement of the ASM vehicle. This is defined in the maneuver.

Maneuver	<p>Maneuvers are predefined sequences of driving instructions for simulating a variety of driving situations with the ASM vehicle. Typical maneuvers are braking on a road with different friction on two lanes (μ-split maneuver), performing a double lane change, and step steering. The maneuver subsystem of the ASM Vehicle Dynamics Simulation Package controls the ASM vehicle, road, and driver models to perform maneuvers.</p>
Maneuver types	<p>There are stimulus maneuvers and controlled maneuvers. The two maneuver types are usually combined in the maneuver segments.</p> <p>Stimulus maneuvers In stimulus maneuvers, the vehicle model is controlled by stimulus signals for accelerator pedal, brake pedal, clutch pedal, gear, and steering wheel. The Scenario Editor makes it easy to enter individual stimulus signals. In addition, you can import stimulus signals from measured data (MAT file).</p> <p>Controlled maneuvers In controlled maneuvers, the driver model controls the vehicle model. Accelerator, brake and clutch pedal positions, and gear are controlled by the driver model's longitudinal controller. The steering wheel is controlled by the lateral controller. For more information on the driver model, refer to Driver (ASM Environment Reference ).</p>
Specifying maneuver segments	<p>The movement of the ASM vehicle is described in sequences that are subdivided into segments arranged in a chain. A segment contains an activity and a transition. An activity specifies the movement of the ASM vehicle in longitudinal and lateral direction and the clutch and gear parameters. A transition specifies when an activity ends and the following activity starts. A transition can have one or more conditions.</p> <p>For details, refer to Basics of Activities on page 31 and Basics of Transitions on page 36.</p>
Specifying initial values	<p>The first segment of a maneuver is used to specify the initial values for the ASM vehicle. You can specify the initial position, initial velocity, and initial vehicle height in this segment. In the corresponding transition, specify a small duration for transient effects, for example, 2 seconds.</p> <p>When the ASM vehicle is to follow a route on the associated road, so you must specify the route on the start segment of each sequence. In addition, you can specify the direction of travel.</p>
Maneuver user signals	<p>To communicate with other elements of the simulation model, user signals are used. Some user signals can be selected in the activity of a maneuver. These are the maneuver user signals.</p>

You can select maneuver user signal and assign a source type to it. The Scenario Editor provides four kind of source types: Constant, ramp, pulse, and table. The selected source type will be set when the corresponding activity is active. It is possible to specify a delay value so that the source type will be set after a specified time.

If a maneuver user signal is specified, the maneuver block on the Scenario pane is marked with a green bar.

Related topics

Basics

[Workflow for Maneuver Creation](#).....40

Basics of Fellows

Introduction

In scenarios, you can simulate the movements of fellows (for example, fellow vehicles or pedestrians) in the environment of the ASM vehicle. The movement of the ASM vehicle is defined in the maneuver sequences. The movements of the fellows are defined in fellow sequences.

Fellow

A definition is the specification of a traffic participant in the environment of an ASM vehicle. Traffic participants can be any kind of object. They can be other vehicles or pedestrians, depending on what traffic scenario is to be simulated.

You can specify the movements of the fellows in absolute values or relative to the position of the ASM vehicle or other fellows.

Geometry

The geometry for the visualization of a fellow is specified by traffic objects. The traffic objects also define the settings for object detection sensors.

Traffic objects can have various geometries. They can be based on vehicles so you can use them for simulating driving situations such as overtaking or parking. Traffic objects can also be based on humans or animals so you can simulated traffic scenarios such as pedestrian crossing.

Note

Do not use grouped objects for fellows. The scene generation fails if fellows have a geometry of grouped objects.

Refer to [Basics of Traffic Objects \(ModelDesk Traffic Object Management !\[\]\(e50091943b385fe16d3277389202856f_img.jpg\)](#)).

Specifying movements

You can specify the movements of each fellow individually. The movements of a fellow are described in sequences that are subdivided into segments arranged in a chain. A segment contains an activity and a transition. An activity specifies the movement of a fellow in longitudinal and lateral direction. A transition specifies when an activity ends and the following activity starts. A transition can have one or more conditions.

Refer to [Basics of Activities](#) on page 31 and [Basics of Transitions](#) on page 36.

Specifying initial values

The first segment of a fellow definition is used to specify initial position for the fellow. You can specify the initial longitudinal and lateral position in this segment. In the corresponding transition, you can specify the **Immediate** condition so that the segment sets only the initial value and has no duration.

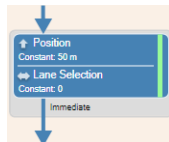
A fellow is to follow a route on the associated road, so you must specify the route on the start segment of each sequence. In addition, you can specify the direction of travel.

Fellow user signals

To communicate with other elements of the simulation model, user signals are used. Some user signals can be selected in the activity of a fellow. These are the fellow user signals.

You can select fellow user signal and assign a source type to it. The **Scenario Editor** provides four kind of source types: Constant, ramp, pulse, and table. The selected source type will be set when the corresponding activity is active. It is possible to specify a delay value so that the source type will be set after a specified time.

If a fellow user signal is specified, the fellow block on the **Traffic Scenario** pane is marked with a green bar, see the following illustration.



Related topics

Basics

[Workflow for Traffic Creation.....](#) 59

Basics of Global User Signals

Introduction	You can specify global user signals independently of the maneuver and fellow definitions.
Global user signals	Global user signals are signals that you can specify independently of the maneuver and fellow definitions. The value of global user signals can change during the execution of the traffic scenario. You can specify different values and the conditions that must be fulfilled to switch from one value to another.
Specifying global user signals	<p>To specify the global user signals, the same mechanism as for maneuver and fellow definitions is used. The global user signals are described in sequences, which are subdivided in segments arranged in a chain. A segment contains an activity and a transition. An activity specifies the source type of the global user signal. The Scenario Editor provides four kind of source types: Constant, ramp, pulse, and table. The selected source type will be set when the corresponding activity is active. A transition specifies when an activity ends and the following activity, which can have another source type for the global user signal, starts. A transition can have one or more conditions.</p> <p>Refer to Basics of Activities on page 31 and Basics of Transitions on page 36.</p>
Related topics	<p>HowTos</p> <p>How to Specify a Global User Signal..... 67</p>

Basics of Activities

Introduction	An activity specifies the movement of the maneuver or a fellow or the value of a global user signal.
Activity for a maneuver	The activity of the maneuver is used to specify its movement. You can specify a longitudinal and lateral profile. The longitudinal profile describes the movement in the direction of travel, such as acceleration or constant velocity. The lateral profile describes the movement at an angle to the direction of travel, such as lane changing. In addition, you can specify the clutch and gear settings of an ASM vehicle with manual transmission or the selector lever settings of an ASM vehicle with automatic transmission.

The settings are used until the transition of the corresponding segment ends the activity and starts the next activity.



Longitudinal profiles You can define different types of stimulation of the ASM vehicle in the longitudinal direction, see the following table.

Longitudinal Type	Description
Velocity	The velocity of the ASM vehicle is specified in absolute values or relative to a fellow. The velocity is constant (as specified by the value in the segment or the value of the previous segment) or changes (as specified by a ramp, a final value, a sine function, a table, or an external value).
Stop	The ASM vehicle brakes until standstill within a specified distance or brakes until it stops. If it brakes until standstill, you can specify a velocity at the beginning of the segment or keep the velocity from the previous segment.
Pedal stimulus	The position of the acceleration and brake pedals are specified. You can specify the position of the pedals using several source types: <ul style="list-style-type: none"> ▪ Specified values (constant, ramp, table) ▪ Manually by the experiment software ▪ Externally by the simulation model
Standstill	The ASM vehicle stands still in this segment.
Lateral acceleration	All the longitudinal definitions of the current segment are ignored. The longitudinal velocity depends on the lateral acceleration. This type can be used only if the lateral type is Basic road and the steering type is Circle .

The parameters to be specified depend on the selected longitudinal type.

Lateral profiles You can define different types of stimulation of the ASM vehicle in the lateral direction, see the following table.

Lateral Type	Description
Not used	The lateral profile is not defined, for example, if the ASM vehicle stands.
Stimulus angle	The lateral profile is specified by an angle of the steering. ModelDesk provides several source types for setting the angle: <ul style="list-style-type: none"> ▪ Specified values (constant, ramp, sine, fixed, step, pulse, sine with dwell, table) ▪ Manually by the experiment software ▪ Externally by the simulation model
Stimulus torque	The lateral profile is specified by a torque at the steering. ModelDesk provides several source types for setting the torque: <ul style="list-style-type: none"> ▪ Specified values (constant, ramp, sine, fixed, step, pulse, sine with dwell, table) ▪ Manually by the experiment software ▪ Externally by the simulation model

Lateral Type	Description
Basic road	The ASM vehicle is driven in a stationary driving situations such as driving straight forward or in a circle at a given lateral acceleration. This type can be used only in scenarios if no road is referenced. For this lateral profile, you should use the Lateral_Control2 subsystem in the simulation model. Refer to Scenario Properties Dialog on page 98 and Lateral Controller 2 (ASM Environment Reference ).
Follow road	The ASM vehicle follows the road on a selected route. For this lateral profile, you should use the Lateral_Control1 subsystem in the simulation model. Refer to Scenario Properties Dialog on page 98 and Lateral Controller 1 (ASM Environment Reference ).

The parameters to be specified depend on the selected lateral type.

Selector lever You can define different types of stimulation of the selector lever of the ASM vehicle with automatic transmission, see the following table.

Selector Lever Type	Description
Driver	The selector lever is specified by the driver model.
Stimulus	The selector lever is specified using source types. ModelDesk provides several source types for setting the selector lever: <ul style="list-style-type: none"> Specified values (constant or table) Manually by the experiment software Externally by the simulation model

Clutch and gear You can define different types of stimulation of the clutch and gear of the ASM vehicle with manual transmission, see the following table.

Clutch and Gear Type	Description
Open clutch	The clutch is open.
Driver	The clutch and the gear is specified by the driver model.
Stimulus	The clutch and the gear is specified using different source types. ModelDesk provides several source types for setting the clutch and gear: <ul style="list-style-type: none"> Specified values (constant, ramp, or table) Manually by the experiment software Externally by the simulation model
Reference gear	The gear is specified using different source types. ModelDesk provides several source types for setting the gear: <ul style="list-style-type: none"> Specified values (constant or table) Manually by the experiment software Externally by the simulation model

Activity for a fellow definition

The activity of a fellow is used to specify their movement. You can specify a longitudinal and lateral profile. The longitudinal profile describes the movement in the direction of travel, such as acceleration or constant velocity. The lateral profile describes the movement at an angle to the direction of travel, such as lane changing.

The profiles are used until the transition of the corresponding segment ends the activity and starts the next activity.

Longitudinal profiles You can define several types of movement of a fellow in the longitudinal direction, see the following table.

Longitudinal Type	Description
Not used	The fellow is not used in this segment.
Continue	The fellow continues the last valid segment definition.
Acceleration	The fellow accelerates in longitudinal direction. The acceleration is constant or changes (as specified by a ramp or a table).
Distance [m]	The longitudinal position is specified by the distance of the fellow to the ASM vehicle or another fellow. You can specify where the reference points of the objects are (main point, front, rear, left, right). The distance is constant or changes (as specified by a ramp, a final value, an external value, or a table).
Distance [s]	The longitudinal position is specified by the time gap of the fellow to the ASM vehicle or another fellow. The time gap is constant or changes (as specified by a ramp, a final value, an external value, or a table).
Position	The longitudinal position is specified by the distance of the fellow to the road start position. The distance is constant (a specified value in the segment or the value of the previous segment) or changes (as specified by a ramp, an external value, or a table).
Velocity	The velocity of the fellow is specified in absolute values or relative to the ASM vehicle or another fellow. The velocity is constant (as specified by the value in the segment or the value of the previous segment) or changes (as specified by a ramp, a final value, a sine function, an external value, or a table).
Stop	The fellow brakes until standstill within a specified distance. You can specify a velocity at the beginning of the segment or keep the velocity from the previous segment.

The parameters to be specified depend on the selected longitudinal type.

Lateral profiles You can define several types of movement of a fellow in the lateral direction, see the following table.

Lateral Type	Description
Not used	The fellow is not used in this segment.
Continue	The fellow continues the last valid segment definition.
Lateral deviation	The fellow's lateral position is defined as a fixed distance to the road's reference line by absolute values or relative to the ASM vehicle or another fellow. The lateral position is constant (a specified value in the segment or the value of the previous segment) or changes (as specified by a ramp, a smooth lane change, a final value, an external value, or a table).
Lane selection	The fellow's lateral position is defined relative to the lanes of the road or relative to the ASM vehicle or another fellow. The lateral position is constant (a specified lane index or the value of the previous segment) or changes (a lane change, a smooth lane change, a final value, an external value, or a table). You can enable a force-to-road option that ensures that the fellow drives on the road even if the specified lane is outside the road. If the lane index is 0, the preferred lane of the direction of travel is used.

The parameters to be specified depend on the selected lateral type.

Activity for a global user signal

The activity of a global user signal is used to specify a value. You can select one of four source types that calculates the value. So the value is constant or changes during the activity.

This source type is valid until the transition of the corresponding segment ends the activity and starts the next activity.

Source type You can use four different source types, see the following table.

Source Type	Description
Constant	The value of the signal is constant.
Ramp	The value of the signal changes with a ramp function.
Pulse	The value of the signal changes with a pulse function.
Table	The value of the signal is read from a table. The values of the table can be imported from a MAT file.

Related topics

References

Activity Properties (Fellow).....	119
Activity Properties (Global User Signal).....	120
Activity Properties (Maneuver).....	121

Basics of Transitions

Introduction

Transitions specify the conditions that end one activity and start the following activity.

Basics

A segment contains a transition and an activity. You can specify one or more conditions for the transition that must be met to end the segment. The first condition to be fulfilled ends the segment.

Conditions

A transition can have various kinds of conditions. The transition to the following segment can be delayed. The following conditions for segment transitions are available.

Duration The segment ends after a specified time.

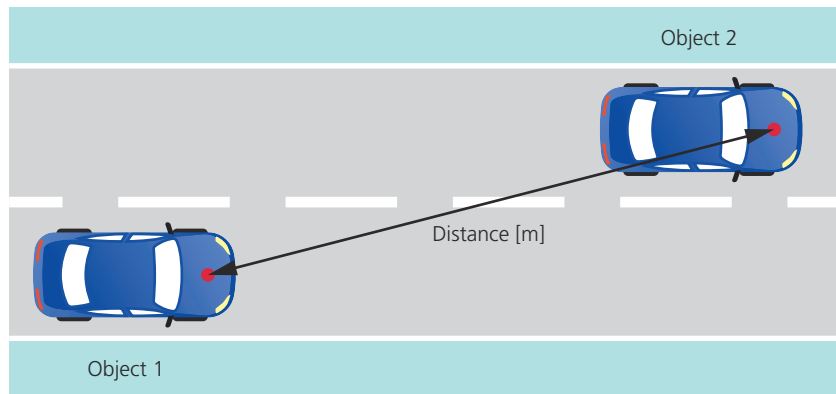
Endless The segment never ends.

Immediate The segment ends immediately. You can use this condition to use a segment for initialization.

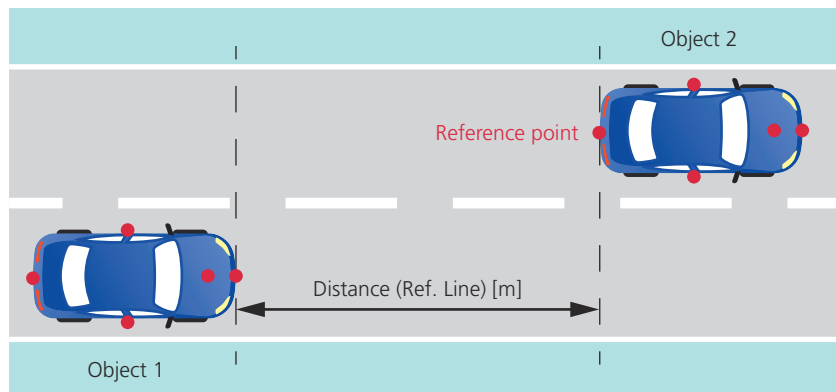
Position The segment ends when the ASM vehicle or fellow reaches a specified position.

Driven Distance The segment ends when the ASM vehicle or fellow traveled the specified distance.

Distance (between main points) The segment ends when the distance between the main points of two objects (ASM vehicle, fellow, or position marker) is greater or smaller than a specified value.



Distance (along road reference line) The segment ends when the distance between two objects (ASM vehicle, fellow, or position marker) is greater or smaller than a specified value. The distance is measured between selectable reference points of the objects along the reference line.



Lateral acceleration (Only for maneuvers) The segment ends when the lateral acceleration of the ASM vehicle is greater or smaller than a specified value.

Time Gap (Only for fellows and global user signals) The segment ends when the time gap between two objects (ASM vehicle, fellow, or position marker) is greater or smaller than a specified value.

Velocity The segment ends when the velocity of the ASM vehicle or fellow is greater or smaller than a specified value.

Trigger The segment ends when it is externally triggered.

Related topics

HowTos

How to Initialize and Specify the Movements of a Fellow.....	63
How to Specify a Transition for a Maneuver.....	53

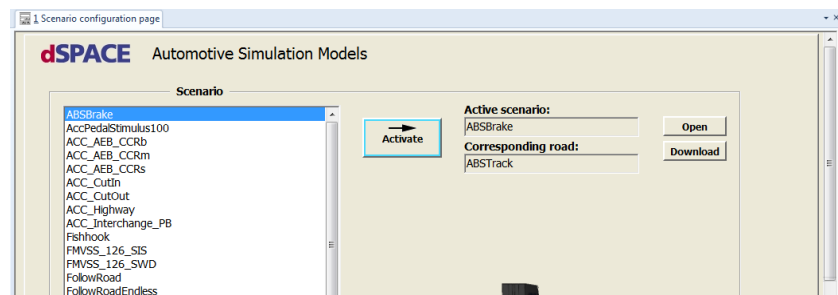
Example of Scenarios with Roads

Introduction

A scenario can be combined with a road. Only if it combined with the correct road, the simulation results make sense.

Combination examples

A scenario usually is related to a road. When you activate a scenario, the related road is displayed on the **Scenario configuration** page. The following illustration shows the example of the **ABSBrake** scenario that is related to the **ABSTrack** road.



Related topics

References

[Open from Pool.....](#) 87

Creating Maneuvers

Introduction

A maneuver specifies the movement of the ASM vehicle.

Where to go from here

Information in this section

Workflow for Maneuver Creation	40
This overview shows the workflow for creating a maneuver.	
How to Start Scenario Creation	41
A maneuver specifies the movement of the ASM vehicle. It is specified with the Scenario Editor.	
How to Add Sequences and Segments for the Maneuver	43
A maneuver specifies the stimulation of the ASM vehicle. The maneuver can consist of one or more sequences. A sequence consist of one or more segments. A segment consists of one activity and one transition.	
How to Accelerate and Brake in a Maneuver	45
You can control the vehicle's accelerating and braking behavior by defining stimulus signals for accelerator and brake pedal or by defining a reference velocity profile.	
How to Shift Gears in a Maneuver	46
You can control the ASM vehicle's gear shifting for manual transmission by specifying clutch and gear, and automatic transmission by specifying the selector lever.	
How to Follow a Road Automatically	48
For a maneuver on a road network, you can specify road-following in your maneuver. So it is not necessary to define the steering.	
How to Specify a Lane Change Maneuver	50
You can specify a lane change maneuver when the ASM vehicle follows a road with multiple lanes.	
How to Steer in a Maneuver	52
You can specify one of the stimulus signals provided, choose a predefined steering profile, or fix the steering wheel.	
How to Specify a Transition for a Maneuver	53
A transition specifies the condition when an activity ends and the succeeding activity begins.	
How to Specify the Properties of the Driver Model	54
The steering wheel, the accelerator and brake pedals, and the clutch and gear are controlled by a driver model, which is specified by a driver type. In ModelDesk, you can specify the parameters for the driver model.	
How to Select and Specify Maneuver User Signals	56
You can define signals at any point or time in the maneuver segment for use as user signals in the Simulink model.	

Workflow for Maneuver Creation

Introduction

This overview shows the workflow for creating a maneuver.

Workflow

A maneuver is necessary if you want to specify the movement of the ASM vehicle. To create a maneuver, you must perform certain steps.

1. You use the Scenario Editor to create a new scenario. Refer to [How to Start Scenario Creation](#) on page 41
2. A maneuver is structured in sequences and segments. Refer to [How to Add Sequences and Segments for the Maneuver](#) on page 43
3. In an activity of a segment, you specify the signals for stimulating the ASM vehicle.
 - To accelerate or brake, you can stimulate the acceleration pedal and brake pedal. Refer to [How to Accelerate and Brake in a Maneuver](#) on page 45.
 - To shift the gears, you can stimulate signals for automatic and manual transmission systems. Refer to [How to Shift Gears in a Maneuver](#) on page 46.
 - To steer the ASM vehicle, you can stimulate the steering wheel by specifying an angle or torque of the steering wheel. Refer to [How to Steer in a Maneuver](#) on page 52.

When the ASM vehicle is to follow the road, stimulating the steering wheel is not necessary. Refer to [How to Follow a Road Automatically](#) on page 48.
4. In the transition of a segment, you define when an activity ends and the next one starts. Refer to [How to Specify a Transition for a Maneuver](#) on page 53.
5. You can specify the parameters of the driver model that controls the ASM vehicle. Refer to [How to Specify the Properties of the Driver Model](#) on page 54.
6. To communicate with the simulation model, you can use maneuver user signals. Refer to [How to Select and Specify Maneuver User Signals](#) on page 56.
7. If you want to simulate traffic situation using ASM Traffic, you can specify the movements of the traffic participants (fellows). Refer to [Creating Traffic of Scenarios](#) on page 59.
8. If the scenario is complete, you can download the parameters for simulation. Refer to [Using Scenarios](#) on page 71.

Related topics

Basics

Basics of the Scenario Editor	20
Workflow for Traffic Creation	59

How to Start Scenario Creation

Objective

A maneuver specifies the movement of the ASM vehicle. It is specified with the Scenario Editor.

This instruction shows the following steps:

1. Creating a new scenario.
2. Specifying the global properties of a scenario, for example, selecting a road.
3. Specifying the start properties of a maneuver, for example, route selection, initial vehicle position and velocity.
4. Specifying a name for the scenario.

Preconditions

A project must be open in ModelDesk.

Method

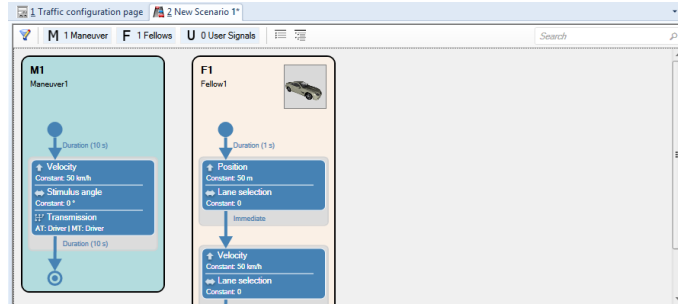
To start scenario creation

- 1 On the Environment ribbon, click Scenario – New.

Or

In the Project Navigator, open the context menu of Scenario and select New.

ModelDesk creates a new scenario and opens the Scenario pane.

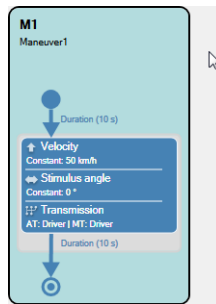


If the ModelDesk experiment is connected to an ASM Traffic Model, the scenario has one Maneuver block and one Fellow block by default. If the ModelDesk experiment is connected to an ASM Vehicle Dynamics Model, the scenario has only the Maneuver block.

You can add more Fellow blocks and Global User Signal blocks. If you want to specify a maneuver only, you can delete the Fellow block.

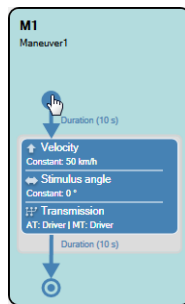
- 2 To delete the Fellow block, select the block, open its context menu and select Delete Fellow.

- 3 To specify the properties of the scenario, click on a free area of the Scenario pane.



The Properties pane displays the scenario properties.

- 4 Specify the properties of the scenario. For details, refer to [Traffic Scenario Properties](#) on page 166.
- 5 To specify the start properties, click the start in the Maneuver block.



The Properties pane displays the start properties.

- 6 Specify the start properties. For details, refer to [Start Properties \(Maneuver\)](#) on page 164.
- 7 To specify the scenario properties, for example, driver properties and controller type, go to the Environment ribbon and click Scenario – Properties.
The Scenario Properties dialog opens.
- 8 Open the Maneuver page, and specify the properties . For details, refer to [Scenario Properties Dialog](#) on page 98.
- 9 The scenario name is specified when you save it:
In the Project Navigator, open the context menu of Scenario and select Save As.
The Specify File Name opens.
- 10 Specify a file name and click Save.

Result

A new scenario is created and the name is specified.

The scenario can be edited, but it is not linked to the current experiment. If you want to link it, use the Activate command.

Next Step

You can specify the maneuver:

- [How to Accelerate and Brake in a Maneuver](#) on page 45
- [How to Shift Gears in a Maneuver](#) on page 46
- [How to Follow a Road Automatically](#) on page 48
- [How to Steer in a Maneuver](#) on page 52
- [How to Select and Specify Maneuver User Signals](#) on page 56

You can specify the traffic of the scenario, refer to [Workflow for Traffic Creation](#) on page 59.

Related topics**Basics**

Basics of Maneuvers.....	27
Scenarios.....	26

References

Activate.....	81
Delete <Element>.....	108
New.....	85
Save As.....	89
Start Properties (Maneuver).....	164
Traffic Scenario Properties.....	166

How to Add Sequences and Segments for the Maneuver

Objective

A maneuver specifies the stimulation of the ASM vehicle. The maneuver can consist of one or more sequences. A sequence consist of one or more segments. A segment consists of one activity and one transition.

Preconditions

A scenario with a maneuver block must be created and open in the Scenario pane, refer to [How to Start Scenario Creation](#) on page 41.

Segment and sequence

Segment To specify several activities of a maneuver that are executed one after the other, you use multiple segments.

You can append a after the last segment or insert a new segment between existing segments. Refer to [Part 1](#) on page 44.

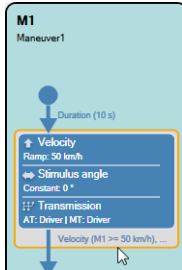
Sequence Using additional sequences, you can specify different initial values. So you can specify different vehicle positions and initial velocities, for example. You can also assign each sequence to a specific route. However, you can only select one road network for the active scenario.

You can only append a sequence after the last existing sequence. Refer to [Part 2](#) on page 44.

Part 1

To add a segment to the maneuver

- 1 Open the context menu of an existing sequence.



- 2 Select one of the following commands:

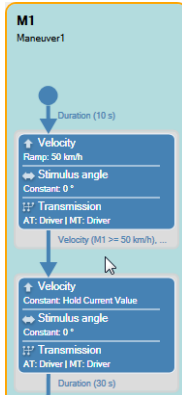
- Append Segment
- Insert Segment – Before
- Insert Segment – After

A segment is inserted.

Part 2

To add a sequence to the maneuver

- 1 Open the context menu of the maneuver block.



- 2 Select Append Sequence.

Result

A sequence and/or segment is appended.

You can specify the behavior in the activities of the maneuver, refer to

- [How to Accelerate and Brake in a Maneuver](#) on page 45
- [How to Shift Gears in a Maneuver](#) on page 46
- [How to Follow a Road Automatically](#) on page 48
- [How to Steer in a Maneuver](#) on page 52
- [How to Select and Specify Maneuver User Signals](#) on page 56

You can specify the transition, refer to [How to Specify a Transition for a Maneuver](#) on page 53

Related topics

Basics

[Basics of Maneuvers.....](#) 27

References

[Append Segment.....](#) 104
[Append Sequence.....](#) 105
[Insert Segment - After.....](#) 110
[Insert Segment - Before.....](#) 111

How to Accelerate and Brake in a Maneuver

Objective

You can control the vehicle's accelerating and braking behavior by defining stimulus signals for accelerator and brake pedal or by defining a reference velocity profile.

Basics of acceleration and braking

Acceleration and braking Acceleration and braking is specified using the longitudinal properties of a maneuver's activity.

- You can control the accelerator and brake pedal by individual stimulus signals. ModelDesk provides simplified profiles (constant, ramp, table). Profiles specified in a MAT file can be imported via a table. It is also possible to stimulate the pedals using the Simulink model or via an experiment software.
- You can specify a reference velocity. ModelDesk provides several profiles (constant, ramp, sine, table). Profiles specified in a MAT file can be imported via a table. It is also possible to set the reference velocity using the Simulink model or via an experiment software.
- You can specify a simplified brake profile.
- You can specify a standstill.

Hold current value You can specify whether the position of the acceleration pedal or brake pedal reached in the previous maneuver segment is held for the current maneuver segment or not by enabling the **Hold current value** option (available only for Constant pedal source). If enabled, you cannot specify values for the current maneuver segment.

Start with current value You can specify whether the position of the acceleration pedal or brake pedal reached in the previous maneuver segment is used as the start value or not by enabling the **Start with current value** option (available only for Ramp pedal source). If enabled, you cannot specify a start value for the current maneuver segment.

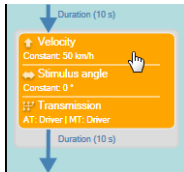
Preconditions

A scenario with a Maneuver block must be created and the Scenario pane must be open, refer to [How to Start Scenario Creation](#) on page 41.

Method

To accelerate and brake in a maneuver

- 1 Select the activity of the maneuver which longitudinal properties you want to specify.



The Properties pane displays the properties of the selected activity. For specifying the acceleration and braking of the ASM vehicle, you must specify the properties of the Longitudinal category. For details, refer to [Longitudinal Type Properties \(Maneuver\)](#) on page 151.

- 2 In the Longitudinal category, select a longitudinal type.
Now, the properties are displayed in the category that are required by the selected type.
- 3 Specify the properties of the selected type.

Result

You specified the acceleration and brake of the ASM vehicle.

Related topics

HowTos

[How to Shift Gears in a Maneuver](#)..... 46

References

[Longitudinal Type Properties \(Maneuver\)](#)..... 151
[Maneuver Scheduler \(ASM Environment Reference !\[\]\(b538fe54c1f3a7343e37e85cc2d00497_img.jpg\)](#))

How to Shift Gears in a Maneuver

Objective

You can control the ASM vehicle's gear shifting for manual transmission by specifying clutch and gear, and automatic transmission by specifying the selector lever.

Basics of shifting gears

For manual transmission, the clutch and gear are specified in the **Clutch and gear** category of a maneuver's activity. For automatic transmission, the selector lever is specified in the **Selector lever** category of a maneuver's activity.

- You can specify the clutch, gear, and selector lever individually by stimulus signals. ModelDesk provides simplified profiles (constant, table, ramp (only for the clutch)). Profiles specified in a MAT file can be imported via a table. It is also possible to stimulate the clutch and gear using the Simulink model or via an experiment software.
- You can also let the driver model of ASM control the gear shifting by selecting the driver mode in the pedal stimulus or desired velocity profile. If you select the final velocity or brake profile (with open clutch disabled) for the longitudinal type, the driver model controls the gear shifting automatically.
- You can specify the gear individually by stimulus signals. ModelDesk provides simplified profiles (constant and table). Profiles specified in a MAT file can be imported via a table. It is also possible to stimulate the gear using the Simulink model or via an experiment software.
- You can open the clutch.

Hold current value You can specify whether the clutch position or gear reached in the previous maneuver segment is held for the current maneuver segment or not by enabling the **Hold current value** option (only available for Constant clutch pedal or gear source). If enabled, you cannot specify values for the current maneuver segment.

Start with current value You can specify whether the clutch position reached in the previous maneuver segment is used as the start value or not by enabling the **Start with current value** option (available only for Ramp clutch pedal source). If enabled, you cannot specify a start value for the current maneuver segment.

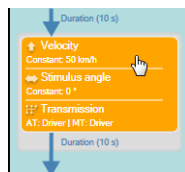
Preconditions

A scenario with a Maneuver block must be created and the Scenario pane must be open, refer to [How to Start Scenario Creation](#) on page 41.

Method

To shift gears in a maneuver

- 1 Select the activity of the maneuver which gear shifting properties you want to specify.



The **Properties** pane displays the properties of the selected activity. For specifying the acceleration and braking of the ASM vehicle, you must specify the properties of the **Clutch and gear** category (manual transmission) or **Selector lever** category (automatic transmission). For details, refer to [Clutch and Gear \(MT\) Properties](#) on page 122 or [Selector Lever \(AT\) Properties](#) on page 161.

- 2 In the **Clutch and gear** or **Selector lever** category, select the **Clutch and gear type** or **Selector lever type**.



The properties that are displayed in the category depends on the selected type.

- 3 Specify the properties of the selected type.

Result You specified the gear shifting.

Related topics

References

Clutch and Gear (MT) Properties	122
Lateral Controller 1 (ASM Environment Reference )	
Lateral Controller 2 (ASM Environment Reference )	
Selector Lever (AT) Properties	161

How to Follow a Road Automatically

Objective For a maneuver on a road network, you can specify road-following in your maneuver. So it is not necessary to define the steering.

Basics of following the road automatically

If the maneuver follows the road automatically, you do not have to define any steering behavior for the vehicle model, because the driver model controls the steering wheel. You can also adapt the velocity to the road by selecting **Adapt velocity to road** as driver type in the **Lateral** category. Then the driver model will drive the vehicle using preview information for road-following and speed adjustment. For information on specifying the global driver settings, refer to [How to Specify the Properties of the Driver Model](#) on page 54.

Lane change maneuver

A road can have several lanes that the ASM vehicle can use for driving. If the ASM vehicle drives on lane 0, it uses the preferred lane in the specified direction of travel. You can change the lane within an activity when the maneuver follows a route on a road network.

When you select the **Force to road** property, you can specify that the ASM vehicle stays on the road even if the specified final lane index is outside the road.

Selecting a route

If a road network consists of several road elements connected by junction elements, a route defines the sequence of road elements that a vehicle can use. You can specify the routes that can be used on a road network in the Road Generator. For details, refer to [How to Specify Routes on Road Networks \(ModelDesk Road Creation !\[\]\(919a2cb85b99741a73c0c31a427236a8_img.jpg\)\)](#).

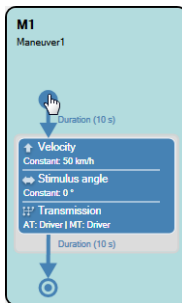
If your maneuver is to follow a route, you must select the route in the Start segment of the maneuver. Refer to [How to Start Scenario Creation](#) on page 41.

Preconditions

A scenario with a Maneuver block must be created and the Scenario pane must be open, refer to [How to Start Scenario Creation](#) on page 41.

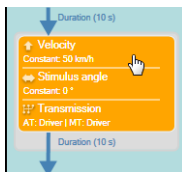
Method**To follow a road automatically**

- 1 Select the Start of the maneuver.



The Properties pane displays the properties of the Start activity.

- 2 In the Start category, select the route for the ASM vehicle.
- 3 Select the activity of the maneuver.



The Properties pane displays the properties of the selected activity.

- 4 In the Lateral type of the Lateral category, select **Follow road**.
- 5 Specify the other properties of the Lateral category. For details, refer to [Lateral Type Properties \(Maneuver\)](#) on page 133.

Result

The ASM vehicle drives along the related route.

Next steps

To specify a lane change maneuver, refer to [How to Specify a Lane Change Maneuver](#) on page 50.

Related topics

HowTos

[How to Steer in a Maneuver.....](#) 52

References

[Lateral Type Properties \(Maneuver\).....](#) 133
[Start Properties \(Maneuver\).....](#) 164

How to Specify a Lane Change Maneuver

Introduction

You can specify a lane change maneuver when the ASM vehicle follows a road with multiple lanes.

Basics of lane change

When an activity of a maneuver has the **Follow road** lateral type, the ASM vehicle drives on the road while being controlled by a controller model. It is therefore not necessary to specify the steering via the properties. However, the ASM vehicle can change lane, for example, to simulate overtaking.

Possible methods

In the Scenario Editor, you can specify the index of the lane to which the ASM vehicle changes in three ways:

- Lane index in relation to the center lane

A lane section of a road element has always one center lane with the lane index 0. The lanes right to the center lane in the direction of travel have negative indices (-1, -2, ...). The lanes left to the center lane have positive indices (1, 2, ...). The lane index can be one of these lane indices. Refer to [Method 1](#) on page 51.

- Lane index in relation to preferred lane

In the lane section of a road element, one lane is marked as the preferred lane for the direct direction. You can specify a lane index in relation to this lane. Specifying positive numbers steers the ASM vehicle to a lane to the left in the direction of travel. Refer to [Method 2](#) on page 51.

- Lane index in relation to the ASM vehicle

You can specify a lane relative to the lane that is used by the ASM vehicle. Specifying positive numbers steers the ASM vehicle to a lane to the left in the direction of travel. Refer to [Method 3](#) on page 51.

Preconditions

The maneuver has an activity with a **Follow road** lateral type. Refer to [How to Follow a Road Automatically](#) on page 48.

Method 1**To specify a lane change maneuver with a lane in relation to the center lane**

- 1** In the Environment ribbon, click Scenario – Properties to open the Scenario Properties dialog.
- 2** On the Global Properties page of the dialog, select Relative to center lane (absolute lane index).
- 3** Click the activity block of the maneuver.
The Properties pane displays the properties of the maneuver's activity.
- 4** In the Lateral category, set Reference type to Relative to lane.
- 5** In the Parameters category, specify the lane index.

Method 2**To specify a lane change maneuver with a lane in relation to the preferred lane**

- 1** In the Environment ribbon, click Scenario – Properties to open the Scenario Properties dialog.
- 2** On the Global Properties page of the dialog, select Relative to preferred lane and driving direction.
- 3** Click the activity block of the maneuver.
The Properties pane displays the properties of the maneuver's activity.
- 4** In the Lateral category, set Reference type to Relative to lane.
- 5** In the Parameters category, specify the lane index.

Method 3**To specify a lane change maneuver relative to the lane used by the ASM vehicle**

- 1** Click the activity block of the maneuver.
The Properties pane displays the properties of the maneuver's activity.
- 2** In the Lateral category, set Reference type to Relative to object.
- 3** In the Parameters category, specify the lane index (number of lanes to be changed).

Result

The lane to which the ASM vehicle is changes is specified.

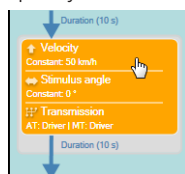
Related topics**References**

Lateral Type Properties (Maneuver)	133
Scenario Properties Dialog	98

How to Steer in a Maneuver

Objective	You can specify one of the stimulus signals provided, choose a predefined steering profile, or fix the steering wheel.
Basics of steering	<p>Steering of the ASM vehicle is specified using the lateral properties of a maneuver's activity.</p> <ul style="list-style-type: none"> You can specify circular or straight activities to make the ASM vehicle drive on them automatically. You can control the steering angle by individual stimulus signals. ModelDesk provides simplified profiles (constant, ramp, sine, sine with dwell, step, pulse, table). Profiles specified in a MAT file can be imported via a table. It is also possible to stimulate the steering angle using the Simulink model or via an experiment software. You can control the steering torque by individual stimulus signals. ModelDesk provides simplified profiles (constant, ramp, table). Profiles specified in a MAT file can be imported via a table. It is also possible to stimulate the steering angle using the Simulink model or via an experiment software. You can hold the current steering wheel angle in the maneuver segment. You can also follow the road. Then, the driver model specifies the steering. Refer to How to Follow a Road Automatically on page 48. <p>Hold current value You can specify whether the steering wheel position reached in the previous maneuver segment is held for the current maneuver segment or not by enabling the Hold current value option (only available for Constant steering source). If enabled, you cannot specify values for the current maneuver segment.</p> <p>Start with current value You can specify whether the steering wheel reached in the previous maneuver segment is used as the start value or not by enabling the Start with current value option (available only for Ramp steering source). If enabled, you cannot specify a start value for the current maneuver segment.</p>
Preconditions	A scenario with a Maneuver block must be created and the Scenario pane must be open, refer to How to Start Scenario Creation on page 41.

Method	<p>To steer in a maneuver</p> <ol style="list-style-type: none"> Select the activity of the maneuver which longitudinal properties you want to specify.
---------------	---



The **Properties** pane displays the properties of the selected activity. For specifying the steering of the ASM vehicle, you must specify the properties of the **Lateral** category. For details, refer to [Lateral Type Properties \(Maneuver\)](#) on page 133.

2 In the **Lateral** category, select a lateral type.

Now, the properties are displayed in the category that are required by the selected type.

3 Specify the properties of the selected type.

Result

You have specified the steering of the ASM vehicle.

Related topics

HowTos

[How to Follow a Road Automatically.....](#) 48

References

[Lateral Type Properties \(Maneuver\).....](#) 133

How to Specify a Transition for a Maneuver

Objective

A transition specifies the condition when an activity ends and the succeeding activity begins.

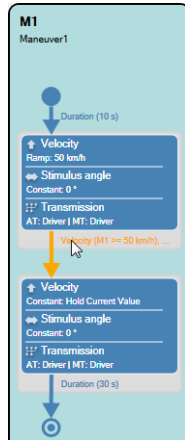
Preconditions

A scenario with a sequence for a maneuver must be created, refer to [How to Add Sequences and Segments for the Maneuver](#) on page 43.

Method

To specify a transition for a maneuver

- 1 In the Scenario pane, click the transition.



The Properties pane displays the properties of the selected transition.

By default, a transition contains one Duration condition. You can remove the default condition and/or add further conditions.

- 2 Add further conditions and specify their properties. Note that some condition types can only be used multiple times if it makes sense. For example, the Duration type can only be used once because multiple different durations are inconsistent.

Result

You specified a transition.

Related topics

Basics

[Basics of Maneuvers.....](#) 27

References

[Transition Properties.....](#) 167

How to Specify the Properties of the Driver Model

Objective

The steering wheel, the accelerator and brake pedals, and the clutch and gear are controlled by a driver model, which is specified by a driver type. In ModelDesk, you can specify the parameters for the driver model.

Basics

ASM contains a driver model that you can select for controlling the ASM vehicle. The driver model support three different driver types and two controller types. For details, refer to [Lateral Controller 1 \(ASM Environment Reference !\[\]\(d84e7ea36f695d92cb39ec32c307ac93_img.jpg\)](#)) and [Lateral Controller 2 \(ASM Environment Reference !\[\]\(db9b0c6fa4ac1078c53d7f74438ad75d_img.jpg\)](#)).

In ModelDesk, you can specify the parameters for the driver model.

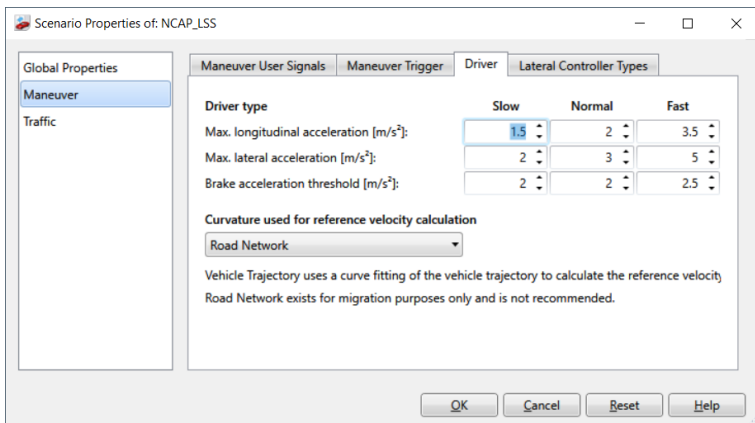
Preconditions

A scenario must be open in ModelDesk.

Method

To specify global driver properties

- 1 On the Environment ribbon, click Scenario – Properties.
The Scenario Properties dialog opens.
- 2 In the dialog, click Maneuver and then Driver.



- 3 Specify the properties.

Result

The properties of the global driver are specified.

Related topics

References

[Lateral Controller 1 \(ASM Environment Reference !\[\]\(f219cfc00b8db0cd1a81ae1fc9afaf28_img.jpg\)](#))
[Lateral Controller 2 \(ASM Environment Reference !\[\]\(681a6e93edfbc41974b9c67a48b93778_img.jpg\)](#))
[Scenario Properties Dialog..... 98](#)

How to Select and Specify Maneuver User Signals

Objective

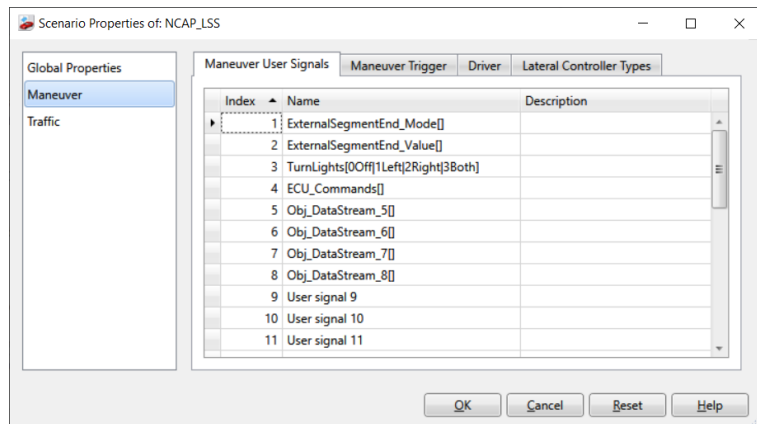
You can define signals at any point or time in the maneuver segment for use as user signals in the Simulink model.

Basics of user signals

When a maneuver segment with user signals runs, signals are generated which can be used in the Simulink model, for example, as triggers. You can access these signals from the ASMSignalBus or from the user signal output of the MANEUVER_SCHEDULER block of the ASM. Refer to [Maneuver Scheduler \(ASM Environment Reference\)](#).

Maneuver user signals

You can create maneuver user signals including a description on the Maneuver User Signals page of the Scenario Properties dialog. The dialog displays the maneuver user signals of the model configuration.



The settings are valid for all maneuver segments that specifies user signals in the current experiment. User signals can be selected in the activity of a maneuver, see below.

Selecting and specifying user signals in the activity

You can select and specify maneuver user signals for each maneuver activity. ModelDesk provides four signal types: Constant, ramp, pulse, and table. You can edit user signal parameters in the Properties pane. Whenever you select a signal type in a user signal row, the appropriate parameters are displayed.

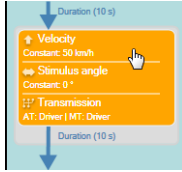
Preconditions

A scenario with a Maneuver block must be created and the Scenario pane must be open, refer to [How to Start Scenario Creation](#) on page 41.

Method

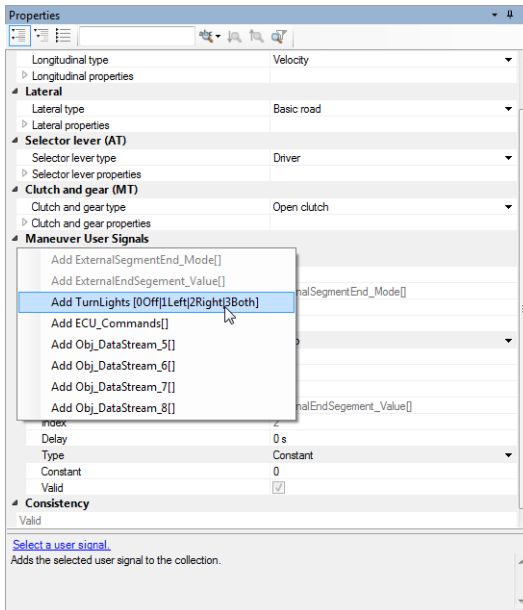
To select and specify maneuver user signals

- 1 Select the activity of the maneuver where you want to add a maneuver user signal.



The Properties pane displays the properties of the selected activity. For selecting and specifying a maneuver user signal, you must specify the properties of the Maneuver User Signals category.

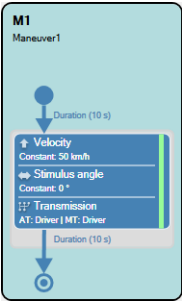
- 2 In the Maneuver User Signals category, click Select a maneuver user signal and select a signal.



- 3 Specify the properties of the maneuver user signal. For details, refer to [Maneuver User Signal Properties](#) on page 159.
- 4 To specify further maneuver user signals, repeat the previous steps.

Result

The maneuver user signals are specified. When you have specified a maneuver user signal in an activity, its block is marked with a green bar.



Related topics

Basics

[ASMSignalBus \(ASM User Guide !\[\]\(a870788d6ed9b8fd294b7654a8c8526b_img.jpg\)](#))

References

Environment Configuration Dialog.....	92
Maneuver User Signal Properties.....	159

Creating Traffic of Scenarios

Introduction

To create a traffic of a scenario, you must specify all the fellows and global user signals.

Where to go from here

Information in this section

[Workflow for Traffic Creation.....](#) 59

This overview shows the workflow for creating a traffic of a scenario.

[How to Create a Fellow User Signal.....](#) 60

Fellow user signals are used to set values of the associated signals in the simulation model.

[How to Create and Specify a Fellow.....](#) 62

Describes how to create a fellow and to select a route and traffic object.

[How to Initialize and Specify the Movements of a Fellow.....](#) 63

The movements of fellows are specified by segments, which contain an activity and a transition.

[How to Specify a Global User Signal.....](#) 67

Describes how to create and specify global user signals to communicate with the simulation model independently of fellows.

[How to Specify the Properties of a Traffic Driver.....](#) 69

You can specify the properties of a traffic driver in ModelDesk.

Information in other sections

[Example of a Scenario \(ASM Traffic Reference \)](#)

The example illustrates the implementation of a scenario in the Scenario Editor and the corresponding signal data from simulation.

Workflow for Traffic Creation

Introduction

This overview shows the workflow for creating a traffic of a scenario.

Workflow for creating a traffic of a scenario

To create a traffic of a scenario, you must specify the fellows and global user signals.

1. Create the scenario. Refer to [How to Start Scenario Creation](#) on page 41.
2. When you create a fellow, you can select a traffic object from the Pool. The traffic object specifies the geometry of the fellow and the 3-D object used in the visualization. Refer to [How to Create and Specify a Fellow](#) on page 62.
3. A new fellow is created with one sequence and two segments by default. The first segment is used to initialize the position of the fellow. The second segment is used to specify its first movement. You can add more segments to the fellow definition to specify different movements. Refer to [How to Initialize and Specify the Movements of a Fellow](#) on page 63.
4. Global user signals are user signals which can be set independently from the fellows. You can create them in the same way as fellow definitions. Refer to [How to Specify a Global User Signal](#) on page 67.

Related topics

Basics

[Scenarios..... 26](#)

How to Create a Fellow User Signal

Objective

Fellow user signals are used to set values of the associated signals in the simulation model.

Fellow user signals

Fellow user signals are used to set values of signals for fellows in the simulation model (for details on the Simulink block, refer to [Traffic Scheduler \(ASM Traffic Reference !\[\]\(758ebdf4629c903da74c2e079717ae32_img.jpg\)](#))). The signals can be used, for example, to set states of fellows. When fellows are visualized in MotionDesk and state objects are used, the turn signals or brake lights can be switched for the fellow. For details, refer to [Basics of Using State Objects in the Scene \(MotionDesk Scene Animation !\[\]\(e7d82ae1e31b23b67694dcc1e3031ff6_img.jpg\)](#)).

Fellow user signals are created in the Scenario dialog. When you specify the activities of fellows, you can select one or more of them and assign values for the states.

Fellow user signals and model configuration

The fellow user signals that you specify in ModelDesk must correspond to the signals in the model configuration. The number and names of fellow user signals in the Scenario dialog must be identical to the fellow user signals specified in the Model Configuration dialog. Otherwise the fellow user signals are invalid.

Preconditions

A scenario must be created and open in the Scenario Editor.

Method

To specify a fellow user signal

1 To open the Scenario dialog, go to the Environment ribbon and click Scenario – Properties.

2 Select the Traffic page.

3 Click on the first row to add a new row.

4 Specify the name according to the name specified in the Model Configuration dialog.

You can also reset the settings of the Scenario Properties dialog. When you do this, all the settings of this dialog are set to the settings of the model configuration in the Model Configuration dialog. This also applies to the fellow user signals.

5 Click Close.

Result

You specified fellow user signals.

Related topics

References

Environment Configuration Dialog.....

Settings.....

92

116

How to Create and Specify a Fellow

Objective

You must select a traffic object and a route for a fellow.

Specifying a fellow

When you want to create a fellow, you have to specify several parts.

Route As a fellow is to follow a route, you must select one that is defined for the selected road network.

Traffic object Select a traffic object to specify the geometry of the fellow. The traffic object defines the parameters for the object detection sensors for the simulation and the 3-D object that will be used to visualize the fellow in MotionDesk.

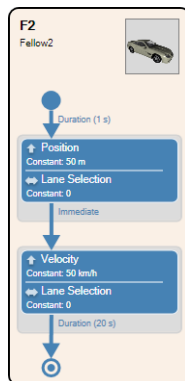
Preconditions

- A scenario must be created and opened in the Scenario Editor.
- The Properties pane must be open.
- The traffic object to be used must be created. Refer to [How to Create a Traffic Object \(ModelDesk Traffic Object Management !\[\]\(c33cb967c8fc4f5e27188a389b621c8e_img.jpg\)](#)).

Method

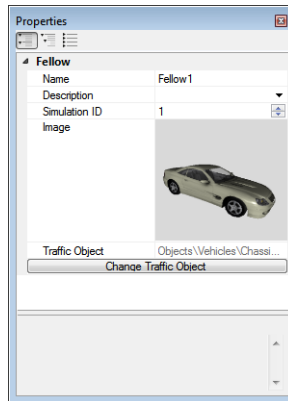
To create and specify a fellow

- 1 On the Scenario pane, open the context menu and select New – Fellow. ModelDesk creates a new fellow with one sequence and two segments.



2 Select the fellow block.

The Properties pane displays the properties of the fellow.



3 Click Change Traffic Object to select a traffic object.

ModelDesk opens the Traffic Object Browser, which displays all the traffic objects of the Pool.

4 Select a suitable traffic object and click OK.

5 Specify the other properties of the fellow. For a description of the properties, refer to [Fellow Properties](#) on page 126.

Result

You specified a fellow.

Related topics

Basics

[Basics of Fellows](#)..... 29

References

[New – Fellow](#)..... 112
[Segment Properties](#)..... 160

How to Initialize and Specify the Movements of a Fellow

Objective

The movements of fellows are specified by segments, which contain an activity and a transition. The specification method is independent of the fellow type. You can specify fellow vehicles in the same way as other traffic participants, such as pedestrians.

Specifying the movement

To specify the segments for a fellow, you have to specify several parts:

Initialization The first segment is used to initialize the fellow. In the activity, you can specify information such as the position of the fellow. In the transition, you can set different types, such as the immediate type. This type ends the segment immediately.

If the first segment is not used for specifying the position, the simulation model automatically sets the fellow's starting position to the origin ($s = 0$ and $d = 0$).

Movement To define the movement of the fellow, you can specify a chain of segments. Each segment contains an activity and a transition.

An activity specifies the movement in the longitudinal and lateral directions. For detailed information on the possible settings, refer to [Basics of Activities](#) on page 31.

A transition specifies the conditions that end an activity. A transition can have one or more conditions of different types. The condition that is fulfilled first ends the segment. For detailed information on the conditions, refer to [Basics of Transitions](#) on page 36.

You can add further segments to specify other activities of the fellow. The segments are executed consecutively.

Selected route When routes are defined on the road network, you must select a route for the fellow to use. In each sequence, the fellow can follow a different route. If you want the fellow to follow other routes, you must create additional sequences.

Fellow user signals Fellow user signals are used to set the values of the associated signals in the simulation model. You can select fellow user signals in the activities. For each fellow user signal, you can specify the value that will be set when the activity of the fellow is executed.

Reusing settings

You can copy settings of an element of a fellow (sequence, segment, activity, or transition) to another element of the same type. You can do this via the Copy and Paste commands of the context menu or via drag & drop.

Preconditions

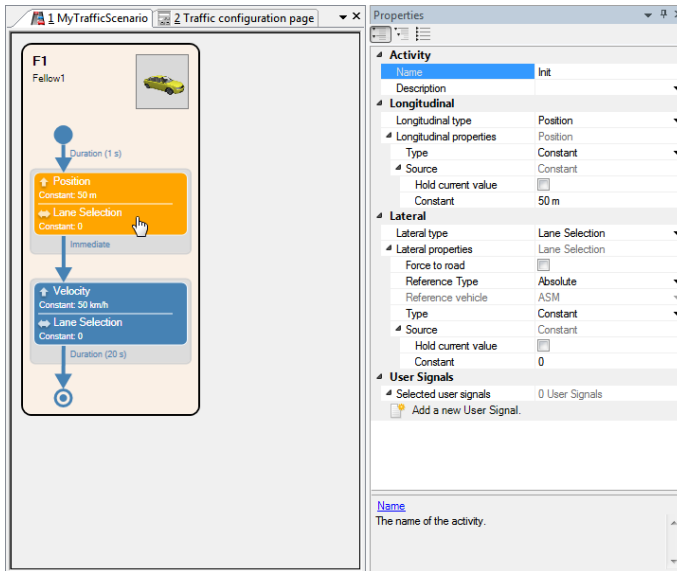
- A scenario must be created and open in the Scenario Editor.
- A road network must be selected for the scenario.
- A fellow must exist (see [How to Create and Specify a Fellow](#) on page 62).
- Fellow user signals must exist (see [How to Create a Fellow User Signal](#) on page 60).
- The Properties pane must be open.

Method

To initialize and specify the movement of a fellow

- 1 On the Scenario pane, select the activity of the first segment.

The Properties pane displays the properties of the activity.



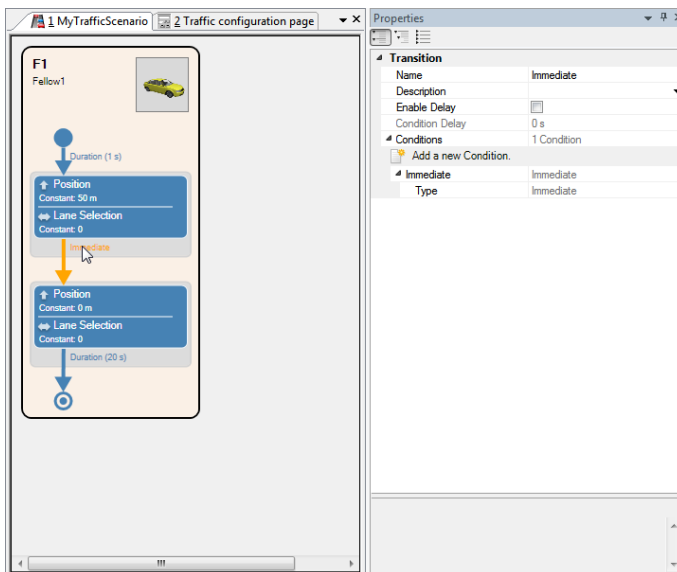
The first segment is used to initialize the position of the fellow.

- 2 In the Longitudinal category, specify a value in longitudinal direction (s coordinate) for the fellow. For details of the properties, refer to [Longitudinal Type Properties \(Fellow\)](#) on page 143.
- 3 In the Lateral category, select a lane or specify a value (d coordinate) for the fellow. For details of the properties, refer to [Lateral Type Properties \(Fellow\)](#) on page 129.

When the position is specified, you can specify the transition.

- 4 On the Scenario pane, select the transition of the first segment.

The Properties pane displays the properties of the transition.



Because the first segment is used to initialize the fellow, the transition has the Immediate condition type by default.

- 5 In the **Conditions** category, specify the conditions that end the activity. You can specify several conditions. The first condition that is fulfilled ends the activity. You should therefore delete the **Immediate** condition if you want to use another condition type. For details of the properties, refer to [Transition Properties](#) on page 167.

Tip

Observe the **Evaluated** property. It shows you whether the appropriate condition will be evaluated in the simulation.

- 6 Click the activity of the second segment. You can modify its properties as desired. For details of the properties, refer to [Longitudinal Type Properties \(Fellow\)](#) on page 143 and [Lateral Type Properties \(Fellow\)](#) on page 129.
- 7 Click the transition of the second segment. You can modify its properties as desired. For details of the properties, refer to [Transition Properties](#) on page 167.
- 8 To add a segment to the sequence, open the context menu of a segment and select
 - **Insert Segment - After** to add the segment after the selected segment
 - **Insert Segment - Before** to add the segment before the current segment
 - **Append Segment** to add the segment after the last segment of the sequence
- 9 Specify the segment, activity, and transition of the new segment as described above.
- 10 To add a new sequence for the fellow, open the context menu of the fellow and select **Append Sequence** on page 105. A further sequence is necessary, for example, when you want to select another route for the same fellow.
- 11 To select a route, select the **Start** element. On the **Properties** pane, you can select the route and direction of travel.
- 12 To select a fellow user signal, select the activity. On the **Properties** pane in the **User Signals** category, select a fellow user signal.

Result

You specified the segments of the fellow.

Related topics


Basics

[Basics of Fellows](#)..... 29

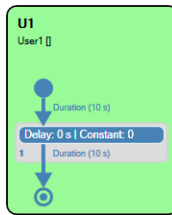
References

[Append Segment](#)..... 104
[Insert Segment - After](#)..... 110
[Insert Segment - Before](#)..... 111
[New – Fellow](#)..... 112

How to Specify a Global User Signal

Objective	You can create and specify global user signals to communicate with the simulation model independently of fellows.
Global user signals	<p>You can use global user signals for communicating with the maneuver scheduler to trigger user-defined actions in the simulation model. Global user signals are independent from the fellows definitions.</p> <p>For details on the maneuver scheduler, refer to Maneuver Scheduler (ASM Environment Reference ).</p>
Specifying global user signals	<p>For specifying global user signals, the same method is used as for specifying fellows. You can specify a chain of segments. Each segment contains an activity and a transition.</p> <p>An activity specifies a constant value for the global user signal.</p> <p>A transition specifies the conditions that end an activity. A transition can have one or more conditions of the same or different types. The condition that is fulfilled first ends the segment. For detailed information on the conditions, refer to Basics of Transitions on page 36.</p>
Reusing settings	<p>You can copy settings of an element of a global user signal (segment, activity, or transition) to another element of the same type. You can do this via the Copy and Paste commands of the context menu or via drag & drop.</p> <p>You can also copy the settings of a fellow's transition and paste them to the transition of a global user signal and vice versa.</p>
Preconditions	<ul style="list-style-type: none"> ▪ A scenario must be created and opened in the Scenario Editor. ▪ A road network must be selected for the scenario. ▪ The Properties pane must be open.
Method	<p>To specify a global user signal</p> <ol style="list-style-type: none"> 1 On the Scenario pane, open the context menu and select New – User Signal.

ModelDesk creates a new global user signal with one segment containing an activity and a transition.



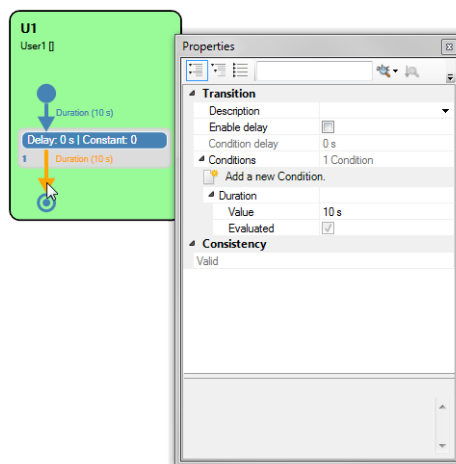
- 2 Select the segment.

The Properties pane displays the properties of the global user signal.

- 3 Specify the properties of the segment. Refer to [Global User Signal Properties](#) on page 129.
- 4 Select the activity of the segment.



- 5 Specify the properties of the activity. Refer to [Activity Properties \(Global User Signal\)](#) on page 120.
- 6 Select the transition of the segment.



- 7 Specify the properties of the transition. Refer to [Transition Properties](#) on page 167.
- 8 To add a segment to the sequence, open the context menu of a segment and select:
 - **Insert Segment - After** to add the segment after the selected segment
 - **Insert Segment - Before** to add the segment before the current segment
 - **Append Segment** to add the segment after the last segment of the sequence
- 9 Specify the segment, activity, and transition of the new segment as described above.

Result You specified a global user signal.

Related topics

References

Append Segment.....	104
Insert Segment - After.....	110
Insert Segment - Before.....	111
Segment Properties.....	160

How to Specify the Properties of a Traffic Driver

Objective You can specify the properties of a traffic driver in ModelDesk.

Basics of traffic driver

The ASM traffic scheduler has the traffic driver feature that provides the following functionalities:

- Fellow vehicles and the ASM vehicle detect other vehicles on the same lane and implement a plausible follow behavior.
- Fellow vehicles and the ASM vehicle detect traffic signs and traffic lights and comply with the corresponding traffic rules.
- Fellow vehicles feature a realistic physical behavior regarding their acceleration and their speed in curves.

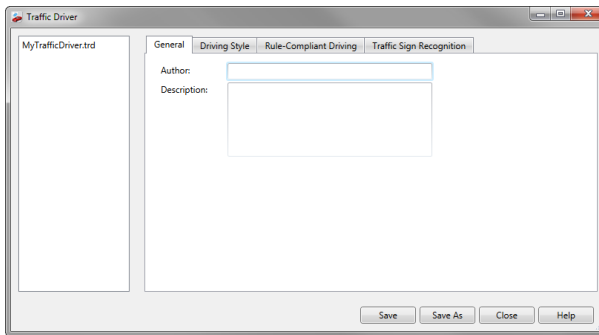
You can specify different driving styles using the traffic drivers in ModelDesk. You can assign the traffic drivers to the vehicles in a scenario. Therefore, different driving styles can be promoted for different vehicles during the simulation.

For details, refer to [Working with the Traffic Driver \(ASM Traffic Guide !\[\]\(bd3b31712ad9bab5a241210fa6925cdd_img.jpg\)](#)).

Method

To specify the properties of a traffic driver

- 1 Open the Traffic Driver dialog:
 - To open an existing traffic driver, go to the Environment ribbon and click Traffic Driver – Open.
 - To create a new traffic driver, go to the Environment ribbon and click Traffic Driver – New.



- 2 Specify the properties of the traffic driver on the pages of the dialog. For details of the properties, refer to [Traffic Driver Dialog](#) on page 101.
- 3 Click Save or Save As to save the properties of the traffic driver.

Result

The properties of the traffic driver are specified.

Related topics

References

Maneuver Properties.....	158
New (Traffic Driver).....	86
Open (Traffic Driver).....	87
Traffic Driver Dialog.....	101





Working with Scenarios

Introduction	You can connect the scenarios you created to roads and work with them in ModelDesk.
--------------	---

Where to go from here	<div>Information in this section</div> <div><div>Using Scenarios..... 71</div><div>You can use scenarios together with roads in the current project or in other parameterization projects. Scenarios can also be downloaded to the simulation model for starting simulation.</div><div>Modifying Properties of a Scenario via Alias Variables..... 72</div><div>When a scenario is created, you can modify properties using alias variables in ModelDesk testing and automation scripts.</div></div>
-----------------------	--

Using Scenarios

Introduction	You can use scenarios together with roads in the current project or in other parameterization projects. Scenarios can also be downloaded to the simulation model for starting simulation.
--------------	---

Using scenarios in other projects	<div>All created scenarios are stored in the Pool, from which you can export them and then import them to the Pool of another project.</div> <div>Export You can export scenario files into a ZIP archive in a similar way to exporting parameter files. Refer to How to Export Parameter Files (ModelDesk Parameterizing ) and Export (ModelDesk Basics ).</div> <div>Import You can import scenario files to your project in a similar way to importing parameter files. Refer to How to Import Parameter Files (ModelDesk Parameterizing ) and Import (ModelDesk Basics ). The scenario files you have imported are saved to the Scenario folder of the Pool automatically. You can link them to your experiment from there.</div>
-----------------------------------	---

Changing parameters outside ModelDesk	<div>You can export a currently linked scenario to change parameters outside ModelDesk.</div> <div>Generate Traffic Ini You must export the scenario INI file to the <code>Simulation\IniFiles\Scenario</code> folder of the ASM to use the scenario together with the model and to run the simulation with. For detailed</div>
---------------------------------------	--

information, refer to [Generate Traffic Ini](#) on page 84. The road is exported as a MAT file.

Downloading scenarios in the simulation model

If you have created a scenario, you must download it to use the specified scenario parameters together with the simulation model. The scenario which is linked to the current experiment can be downloaded to the real-time hardware, Simulink, or VEOS with ModelDesk.

You can download the complete scenario, or the maneuver and traffic content (fellows and global user signals) separately.

Tip

To visualize the traffic fellows in MotionDesk, you must update the scene. Refer to [Basics of Synchronizing the Scene in MotionDesk \(ModelDesk Scene Synchronization !\[\]\(fa6f3af6bfa46c5d4a2d362681095beb_img.jpg\)](#)). The update can be automated by selecting an option. Refer to [Customize \(ModelDesk Scene Synchronization !\[\]\(a9bc825d1a15412853cf9ebcbd72219d_img.jpg\)](#).

Related topics

References

Download / Download Scenario.....	82
Download Maneuver.....	83
Download Traffic.....	83

Modifying Properties of a Scenario via Alias Variables

Introduction

When a scenario is created, you can modify properties using alias variables in ModelDesk testing and automation scripts.

Alias variables

Using alias variables is an easy method to modify properties of the active scenario. The method allows you to modify one or more properties using one alias variable.

You create alias variables in the Scenario Editor for each property that you want to modify. When you create an alias variable, it gets a property reference that points to the property that you want to modify. Then, you can assign several property references to the same alias variable. When you modify the value of the alias variable, you also modify the values of all assigned property references. The modification is done via ModelDesk testing or tool automation.

Related topics

Basics

[Basics of the Alias Support \(ModelDesk Testing 📖\)](#)

[Setting Values of Properties Using Alias Variables \(ModelDesk Automation 📖\)](#)

HowTos

[How to Create Alias Variables \(ModelDesk Testing 📖\)](#)

Controlling Maneuvers and Driving Cycles

Introduction You can use ModelDesk to start, stop, or reset maneuvers for vehicle dynamics test and driving cycles for engine tests.

Where to go from here

Information in this section

Basics of Controlling Maneuvers and Driving Cycles.....	74
You can use ModelDesk to start, stop, and reset maneuvers and driving cycles.	
How to Control a Maneuver or a Driving Cycle.....	76
Provides instructions on how to control (start, stop, reset) a maneuver or driving cycle by using ModelDesk.	

Basics of Controlling Maneuvers and Driving Cycles

Introduction You can use ModelDesk to start, stop, and reset maneuvers and driving cycles.

Basics

When you test ECUs, you want to control maneuvers and driving cycles.

Maneuvers Maneuvers are used to test ECUs for vehicle dynamics. You can specify how a driver drives the ASM vehicle by using the **Scenario Editor**.




Driving cycles Driving cycles are used to test engine control units. The driving cycles are modeled with the ASMs.

ModelDesk can start, stop, or reset the maneuvers/driving cycles only if the simulation application contains the specific ASM blocks. When you use the maneuver control of ModelDesk, ModelDesk writes a pulsed value and a unique number to a parameter of the simulation model. This parameter is evaluated by the specific ASM blocks to control the maneuver.

You can use maneuver control in real-time simulation, Simulink simulation, and offline simulation using VEOS.

Extended simulation model

The simulation application must have a specific code so that ModelDesk can control maneuvers and driving cycles. This code is integrated in the simulation application when you use the **ManeuverControl** blocks of the **ASM_Utils_lib** library. The appropriate commands are enabled only if ModelDesk finds such blocks in the model during a consistency check.

Block	Button	Description
MANEUVER_START		To start the maneuver or driving cycle.
MANEUVER_STOP		To stop the maneuver or driving cycle.
RESET		To reset the states of the simulation application.
MANEUVER_STATE	—	To get the state.

Multiple ManeuverControl blocks

ModelDesk can evaluate only one ManeuverControl block of the same type. If a simulation model has more than one block of the same type, a priority defines which block is evaluated. The priority is defined based on the components as follows:

1. VehicleDynamics
2. Engine
3. ElectricComponents

Supported platform

You can use maneuver control on VEOS, Simulink simulation, and real-time platforms.

Workflow

Some steps must be performed to be able to control maneuvers and driving cycles

1. The simulation model must contain the blocks for maneuver control. Refer to:
 - [Maneuver Scheduler \(ASM Environment Reference !\[\]\(2a133ebb0337313d16cc068f19494aa2_img.jpg\)\)](#) for vehicle dynamics applications
 - [Maneuver Control \(ASM Drivetrain Basic Reference !\[\]\(e5831951c2bb646a242d812c288ddabc_img.jpg\)\)](#) for engine applications
 - [Maneuver and Reset Control \(ASM User Guide !\[\]\(767ddc536c5331f5333c7801240a378b_img.jpg\)\)](#)
2. Register the platform that you want to use for simulation. Refer to [Registering Platforms \(ModelDesk Platform Management !\[\]\(7379045168890876f99aa36845a7ccf9_img.jpg\)\)](#).
3. Create a ModelDesk project and experiment. When you create an experiment, choose the simulation model that contains the ManeuverControl blocks. Refer to [Creating Projects and Experiments \(ModelDesk Project and Experiment Management !\[\]\(42f4a0fde8ff3fc8d2b462e1f7f61ba8_img.jpg\)\)](#).
4. Specify the ModelDesk experiment. If you want to control a maneuver for vehicle dynamics, create a maneuver using the Scenario Editor and/or a road by using the Road Generator.
5. Optional: Download the simulation application and the active experiment to the platform. This step is optional because the Start command can download the simulation application and the active experiment to the platform if no or another simulation application is loaded.
6. Finally, you can control the maneuver or driving cycle. Refer to [How to Control a Maneuver or a Driving Cycle](#) on page 76.

Each time you change the maneuver state, ModelDesk checks whether it is allowed. If it is not allowed, ModelDesk displays an appropriate message in the **Message Viewer**.

Related topics

Basics

[Overview of the Maneuver and Reset Control \(ASM User Guide !\[\]\(e78f798d4ea5c530c9db49e7d26e6b95_img.jpg\)\)](#)

How to Control a Maneuver or a Driving Cycle

Objective

This topic provides instructions on how to control (start, stop, reset) a maneuver or driving cycle by using ModelDesk.

Preconditions

To be able to control the maneuver, the following preconditions must be fulfilled.

- An experiment must be active.
- A platform and a simulation model must be selected.
- The simulation model must contain **ManeuverControl** blocks.

Method

To control a maneuver or a driving cycle

- 1** On the Home ribbon, click **Maneuver Control – Start**.

ModelDesk checks the platform and the running simulation application.

- If no simulation application is running on the platform, you can have ModelDesk download the simulation application and the road, scenario, and parameter set of the active experiment.
- If another simulation application is running on the platform, you can stop the running simulation application and download the simulation application matching the active experiment and the road, scenario, and parameter set of the active experiment.

If the simulation application is running with the parameters of the active experiment and the current maneuver state is suitable, the maneuver starts.

- 2** Check the **Message Viewer**. If ModelDesk was not able to start the maneuver, the **Message Viewer** displays the appropriate message.
- 3** To stop the maneuver, go to the Home ribbon and click **Maneuver Control – Stop**.
- 4** To reset the maneuver of the simulation application, go to the Home ribbon and click **Maneuver Control – Reset**.

Result

You controlled the maneuver.

Related topics

Basics

Basics of Controlling Maneuvers and Driving Cycles.....	74
---	----

References

Reset.....	175
Start.....	176
Stop.....	177

Reference Information

Where to go from here

Information in this section

Scenario Editor Management Commands.....	80
To manage scenarios.	
Scenario Editor Dialogs and Panes.....	91
To specify scenarios.	
Scenario Editor Creation Commands.....	103
To specify scenarios.	
Scenario Editor Properties.....	118
To specify properties of a scenario.	
Maneuver Control Commands.....	175
ModelDesk provides commands for controlling the maneuver.	

Scenario Editor Management Commands

Introduction

ModelDesk provides various commands to manage the scenarios.

Where to go from here


Information in this section

Activate	81
To activate the open scenario by linking it to the current experiment.	
Activate from Pool	81
To activate a scenario saved in the Pool by linking it to the current experiment.	
Download / Download Scenario	82
To download the active scenario to the simulation platform.	
Download Maneuver	83
To download the maneuver of the active scenario to the simulation platform.	
Download Traffic	83
To download the fellows and global user signals of the active scenario to the simulation platform.	
Generate Traffic Ini	84
To export a currently linked scenario to change parameters outside ModelDesk.	
New	85
To create a new scenario.	
New (Traffic Driver)	86
To create a new traffic driver.	
Open	86
To open a scenario linked to the current experiment.	
Open (Traffic Driver)	87
To open a traffic driver stored in the Pool of the ModelDesk project.	
Open from Pool	87
To open a scenario from the Pool.	
Remove	88
To remove a scenario from the current experiment.	
Save	89
To save the scenario.	
Save As	89
To save the current scenario under a new name and path.	

Activate

Access

You can access the command via:

Ribbon	Environment – Scenario
Context menu of	None
Shortcut key	None
Icon	

Purpose

To activate the open scenario by linking it to the current experiment.

Result

The currently open scenario is linked to the experiment and active.

Related topics

Basics

[Using Scenarios..... 71](#)

Activate from Pool

Access

You can access this command via:

Ribbon	None
Context menu of	<ul style="list-style-type: none"> Project Navigator – Scenario Configuration page (scenario)
Shortcut key	None
Icon	None

Purpose

To activate a scenario saved in the Pool by linking it to the current experiment.

Result

ModelDesk opens the Specify File Name dialog for you to select the scenario file from the Pool and activate it by linking it to the current experiment.

Description

To view or edit the active scenario, it can be opened via the Open command.


Related topics**References**

[Open..... 86](#)

Download / Download Scenario

Access

You can access this command via:

Ribbon	<ul style="list-style-type: none"> ▪ Home – Experiment – split button ▪ Environment – Scenario
Context menu of	Project Navigator – Scenario
Shortcut key	None
Icon	
Button	<ul style="list-style-type: none"> ▪ Configuration page (scenario) ▪ Scenario configuration page

Purpose

To download the active scenario to the simulation platform.

Description

A scenario includes the specification of the maneuver, fellows, and global user signals.

Result

The currently active scenario is downloaded.

Description

Only the active scenario can be downloaded. If the open scenario is not the active one, a message dialog appears asking you if you want to activate the open scenario.

In the real-time mode, ModelDesk generates a parameter file, which can be loaded to the real-time hardware. In the Simulink mode, the scenario parameters are downloaded to Simulink for further use.


Related topics**Basics**

[Using Scenarios..... 71](#)

Download Maneuver

Access

You can access this command via:

Ribbon	Environment – Scenario – split button
Context menu of	None
Shortcut key	None
Icon	
Button	None

Purpose

To download the maneuver of the active scenario to the simulation platform.

Result

The maneuver is downloaded.

Related topics


Basics

[Using Scenarios..... 71](#)

Download Traffic

Access

You can access this command via:

Ribbon	Environment – Scenario – split button
Context menu of	None
Shortcut key	None
Icon	
Button	None

Purpose

To download the fellows and global user signals of the active scenario to the simulation platform.

Result The fellows and global user signals are downloaded.

Related topics

Basics

[Using Scenarios..... 71](#)

Generate Traffic Ini

Access

You can access the command via:

Ribbon	None
Context menu of	Project Navigator – Scenario
Shortcut key	None
Icon	None

Purpose

To export a currently linked scenario to change parameters outside ModelDesk.

Result

The scenario is exported to the specified folder and named as a MAT file.

Description

The initialization file of a scenario must be exported to **Simulation\IniFiles\Scenario** of the ASM to use the scenario together with the model and to run the simulation with.

ModelDesk opens the **Generate MAT File** dialog for you to select the path and folder to save the initialization file to.

Save As dialog

Save in Lets you select the path and folder to export the file to.

File name Lets you specify the name of the file. You can also select a name from the files listed under the chosen path and folder.

Save as type Lets you save the file as a MAT file.

If you select a file from those listed under the chosen path and folder, you are prompted before it is overwritten.

Related topics


Basics

Using Scenarios..... 71

New

Access

You can access this command via:

Ribbon	Environment – Scenario
Context menu of	Project Navigator – Scenario
Shortcut key	None
Icon	

Purpose

To create a new scenario.

Description

ModelDesk opens the Scenario pane. The pane has a maneuver and if ASM Traffic Model is used, a fellow by default. You can create further fellows and global user signals in this pane by using the New – Fellow or New – User Signal command.

You can specify the properties of the scenario, such as the related road, in the Properties pane.

The name of the scenario is specified when it is saved.

Related topics

HowTos

How to Start Scenario Creation..... 41

References

New – Fellow..... 112

New – User Signal..... 114

Save As..... 89

Scenario Pane..... 96

Traffic Scenario Properties..... 166

New (Traffic Driver)

Access

You can access this command via:

Ribbon	Environment – Traffic Driver
Context menu of	None
Shortcut key	None
Icon	

Purpose

To create a new traffic driver.

Result

ModelDesk opens the Traffic Driver dialog for you to specify the properties of the traffic driver.

Related topics

HowTos

[How to Specify the Properties of a Traffic Driver.....](#) 69


References

[Traffic Driver Dialog.....](#) 101

Open

Access

You can access this command via:

Ribbon	Environment – Scenario
Context menu of	Project Navigator – Scenario
Shortcut key	None
Icon	

Purpose

To open a scenario linked to the current experiment.

Result ModelDesk opens the linked scenario in the Scenario Editor.

Related topics**Basics**

[Scenarios..... 26](#)

Open (Traffic Driver)

Access

You can access this command via:

Ribbon	Environment – Traffic Driver
Context menu of	None
Shortcut key	None
Icon	

Purpose

To open a traffic driver stored in the pool of the ModelDesk project.

Result

ModelDesk loads the properties of the traffic driver and opens the Traffic Driver dialog.

Related topics**HowTos**

[How to Specify the Properties of a Traffic Driver..... 69](#)

References

[Traffic Driver Dialog..... 101](#)

Open from Pool

Access

You can access this command via:

Ribbon	Environment – Traffic
Context menu of	Project Navigator – Scenario

Shortcut key	None
Icon	

Purpose To open a scenario from the Pool to edit it.

Result ModelDesk opens the standard Specify File Name dialog for you to select the path, folder, file type, and name of the scenario to be opened from the ModelDesk Pool.

Description The scenario can be edited, but it is not linked to the current experiment. If you want to link it, use the **Activate** command.

Related topics

Basics

[Scenarios.....26](#)

References

[Activate.....81](#)

Remove

Access You can access the command via:

Ribbon	None
Context menu of	Project Navigator – Scenario
Shortcut key	None
Icon	None

Purpose To remove a scenario from the current experiment.

Result The active scenario is removed from the current experiment.

Description If no scenario is linked to the current experiment, the command is unavailable.


Related topics**Basics**

[Using Scenarios..... 71](#)

Save

Access

You can access this command via:

Ribbon	Environment – Scenario
Context menu of	Project Navigator – Scenario
Shortcut key	None
Icon	

Purpose

To save the scenario.

Result

The currently open scenario is saved.

Related topics**Basics**

[Workflow for Maneuver Creation..... 40](#)
[Workflow for Traffic Creation..... 59](#)

References

[Save As..... 89](#)

Save As

Access

You can access this command via:

Ribbon	None
Context menu of	Project Navigator – Scenario
Shortcut key	None
Icon	None

Purpose	To save the current scenario under a new name and path.
Result	ModelDesk saves the scenario under a new name and path that you specified.
Description	ModelDesk opens the Specify File Name dialog to change the current entries.
Specify File Name dialog	<div>Save in Lets you select the path and folder to export the file to.</div> <div>File name Lets you specify the name of the file. You can also select a name from the files listed under the chosen path and folder.</div> <div>Save as type Lets you save the file as an XML file. If you select a file from those listed under the chosen path and folder, you are prompted before it is overwritten.</div>
Related topics	<div>Basics</div> <div><div>Workflow for Maneuver Creation.....40</div><div>Workflow for Traffic Creation.....59</div></div> <div>References</div> <div><div>Save.....89</div></div>

Scenario Editor Dialogs and Panes

Introduction The Scenario Editor has several panes that are necessary to specify scenarios.

Where to go from here

Information in this section

Edit Table Dialog.....	91
To specify values of table.	
Environment Configuration Dialog.....	92
To configure settings for the simulation model.	
Scenario Configuration Page.....	94
To activate, open, and download scenarios linked to the current active experiment.	
Scenario Pane.....	96
To specify a scenario.	
Scenario Properties Dialog.....	98
To configure the properties of the open scenario.	
Scenario Settings Dialog.....	100
To configure settings for the simulation model.	
Traffic Driver Dialog.....	101
To configure settings for the simulation model.	

Edit Table Dialog

Access The Edit Table dialog opens when you click the Edit table button for a property.

Purpose To specify values of a table.

Description The table is used for various properties, which can be specified by a map. The x-axis and y-axis of the table are adapted to the property.

To specify the values of a table, you add columns to the table using the Add command of the context menu and modify the values in the table cells. The area above the table displays a curve with the specified values.

To delete unnecessary columns, select and delete them with the Remove command.

It is possible to export and import the values of the table to a MAT file. So you can reuse specified tables for other properties.

Commands

The dialog has buttons and a context menu that provides the following commands.

Add Lets you add a column to the table

Remove Lets you remove the selected column.

Import Lets you load the values of a MAT file to the table. When the MAT file is loaded, existing values are overwritten.

Export Lets you save the values of the table to a MAT file.

Related topics

References

[Scenario Editor Properties..... 118](#)

Environment Configuration Dialog

Access

The dialog opens when you select the Configure command.

Ribbon	Environment – Environment
Context menu of	None
Shortcut key	None
Icon	

Purpose

To configure settings for the simulation model.

Description

In this dialog, you can set certain parameters for the simulation model. Some parameters should be modified only when the simulation model is also modified. Otherwise, the download will fail. To avoid any unintended modification, you must enable the Advanced Configuration option for modifying these parameters.


The dialog has four pages where you can specify parameters for maneuver, road, traffic, and testing, and commands available via buttons and context menu.

Maneuver dialog settings

User Signals Displays the user signals of the maneuver.

Triggers Displays the triggers of the maneuver.

Road dialog settings

Activate OSI Road Support Lets you activate the OSI road support. Activate this option if you want to download a road which data is prepared according to the ground truth data format of the Open Simulation Interface (OSI). For details, refer to [Groundtruth](#) ([ASM Traffic Reference](#) ).

Max Lane Boundary Point Distance Lets you specify the maximum distance of the lane boundary points in meters.

Traffic dialog settings

You can modify the following properties only if the Advanced Configuration option is enabled.

Maximum number of fellows Lets you specify the maximum number of fellows that you want to use in the traffic.

Fellow User Signals Displays the index, name, and description of the fellow user signals.

Global User Signals Displays the index, name, and description of the global user signals.


Triggers Displays the index, name, and description of the triggers.

Testing dialog settings

You can modify the following properties only if the Advanced Configuration option is enabled.

The page lets you specify the signal path and signal collector of the maneuver state and maneuver time signal.

Signal Path Lets you specify the path and name of the signal containing the maneuver state or maneuver time.

Signal Collector Lets you specify the label of the signal collector from which the maneuver state or maneuver time is read. The signal collector is an ASM block that collects signals of the model to provide them to ModelDesk for plotting and testing. For details, refer to [ASMSignalInterface](#) ([ASM User Guide](#) ).

If you modify the label of the `ASMSignalInterface` block in the model, you must adapt this property and update the model in ModelDesk.

If the model is migrated from Release 2020-A or earlier, the label is `Simulation model1`.

Global dialog settings

The dialog has the following buttons.

Import Lets you import the configuration of all pages from an XML file.

Export Lets you export the configuration of all pages to an XML file.

Advanced Configuration Lets you enable or disable the advanced configuration mode.

Note

Some parameters can be modified only in the advanced configuration mode. Modifying these parameters can lead to errors during download or simulation when they are not consistent with the simulation model.

The dialog has a context menu with the following commands.

Move up Lets you move up the selected row.

Move down Lets you move down the selected row.

Delete Lets you delete the selected row.

Related topics

Basics

[Configuring the Traffic Options \(ASM Traffic Guide !\[\]\(d5d7044e5caf6907399af2dced8d6ff8_img.jpg\)\)](#)

HowTos

[How to Create a Fellow User Signal..... 60](#)
[How to Increase the Maximum Number of Fellows \(ASM Traffic Guide !\[\]\(ab4e2b3fc7e7887b7a72f548aa6f5e60_img.jpg\)\)](#)
[How to Increase the Number of Fellow User Signals \(ASM Traffic Guide !\[\]\(0a20d1259d5ab849a22cc9906b421113_img.jpg\)\)](#)
[How to Increase the Number of Global User Signals \(ASM Traffic Guide !\[\]\(7aa29892bff760d52365d7f4c0908c26_img.jpg\)\)](#)
[How to Increase the Number of Trigger Signals \(ASM Traffic Guide !\[\]\(5c1b64962792396e8fbc2d4fb35c2fdb_img.jpg\)\)](#)

References

[OSI Groundtruth Interface \(ASM Traffic Reference !\[\]\(aab88c0d099e5d18d6533a97b13ec28d_img.jpg\)\)](#)

Scenario Configuration Page

Access

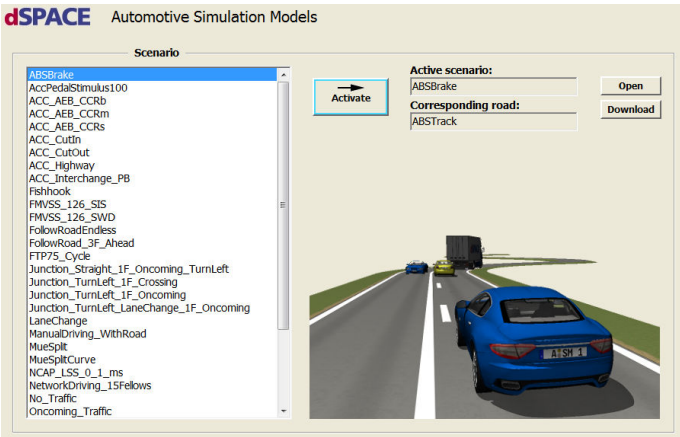
The Scenario Configuration page opens automatically when you select the Scenario node in the Project Navigator.

Purpose

To activate, open, and download scenarios linked to the current active experiment.

Description

The following illustration shows the Scenario Configuration page.



Dialog settings

- Scenario** Lists all the available scenarios saved in the Pool for you to choose one.
- Activate** Lets you activate the selected scenario. Only one scenario can be active in a ModelDesk project at a time. All the others are inactive. To work with a scenario, it must be active.
- Active scenario** Displays the current active scenario.
- Corresponding road** Displays the road name if the current active scenario is linked to a road.
- Open** Opens the Scenario Editor to let you edit the active scenario.
- Download** Lets you download the selected scenario to the model. Refer to [Download / Download Scenario](#) on page 82.

Related topics

References

Activate	81
Download / Download Scenario	82
Open	86

Scenario Pane

Purpose

To specify a scenario.

Description

In the pane, you can specify the maneuver, fellows and global user signals of the scenario.

Commands

The pane displays different elements. When an element is selected, a specific context menu is provided.

No selected element When no element is selected, the following commands are available.

Command	Purpose
New – Maneuver	To add a maneuver to the scenario.
New – Fellow	To add a new fellow to the scenario.
New – User Signal	To add a new global user signal to the scenario.
Layout Diagram	To arrange all the fellows and global user signals on the Scenario pane.

Fellow or global user signal When a maneuver, fellow, or a global user signal is selected, the following commands are available.

Command	Purpose
New – Maneuver	To add a maneuver to the scenario.
New – Fellow	To add a new fellow to the scenario.
New – User Signal	To add a new global user signal to the scenario.
Append Sequence	To add a new sequence to the selected element.
Copy	To copy the element to the Clipboard.
Cut	To cut the selected element and copy it to the Clipboard.
Paste	To paste an element from the Clipboard to the current position.
Delete <element>	To delete the selected element.
Collapse	To collapse the selected block.
Expand	To expand the selected block.

Start segment When a start segment is selected, the following commands are available.

Command	Purpose
Append Segment	To add a new segment at the end of the sequence.
Append Sequence	To add a new sequence to the selected element.
Copy	To copy the element to the Clipboard.

Command	Purpose
Paste	To overwrite the start segment properties with properties of the start segment in the Clipboard.
Delete Sequence	To delete the sequence starting with the selected element.

Activity When an activity is selected, the following commands are available.

Command	Purpose
Copy	To copy the element to the Clipboard.
Paste	To overwrite the element properties with properties of the element in the Clipboard.

Transition When a transition is selected, the following commands are available.

Command	Purpose
Copy	To copy the element to the Clipboard.
Paste	To overwrite the element properties with properties of the element in the Clipboard.

Segment When a segment is selected, the following commands are available.

Command	Purpose
Insert Segment - After	To insert a new segment after the selected segment.
Insert Segment - Before	To insert a new segment before the current segment.
Append Segment	To add a new segment at the end of the current sequence.
Copy	To copy the element to the Clipboard.
Cut	To cut the selected element and copy it to the Clipboard.
Paste	To overwrite the element properties with properties of the element in the Clipboard.
Delete	To delete the selected element.

Related topics

Basics

[Graphical User Interface of the Scenario Editor..... 22](#)

Scenario Properties Dialog

Access	The Scenario Properties dialog opens when you select the Properties command.
Purpose	To configure the properties of the open scenario.
Description	The property values that you specify in this dialog must be consistent to the property values in the Model Configuration dialog. The Scenario Editor checks whether the settings of both dialogs are consistent and shows the inconsistencies. To get consistent values, you can click the Reset button. This sets all the settings of the Scenario Properties dialog to the values of the Model Configuration dialog.

Global Properties

Lane index interpretation Lets you specify which lane is used as reference lane when you select the lane index. For details, refer to [How to Specify a Lane Change Maneuver](#) on page 50.

Property	Description
Relative to preferred lane and driving direction	The reference lane is the preferred lane. Negative lane indices indicate lanes to the right of the preferred lane in relation to the direction of travel of the traffic participant. Positive lane indices indicate lanes to the left.
Relative to center lane (absolute lane index)	The reference lane is the center lane. The center lane is specified with the Road Generator and always has the index 0. The direction of travel of the traffic participants is not considered.

Maneuver dialog settings

Lets you specify the global settings for the maneuver.

Maneuver User Signals Lets you edit a table containing the maneuver user signals. You can add a description to each user signal and modify their names.

Maneuver Trigger Lets you edit a table containing the maneuver trigger signals. You can add a description to each trigger.

Driver Lets you specify driver settings.

Property	Description
Driver type	Lets you specify standard values for a slow, normal, and fast driver. The driver settings are used when the ASM vehicle is to follow the road (refer to Lateral Type Properties (Maneuver) on page 133).

Property	Description
Max. longitudinal acceleration	Lets you specify the maximum acceleration in longitudinal direction in m/s^2 .
Max. lateral acceleration	Lets you specify the maximum acceleration in lateral direction in m/s^2 .
Brake acceleration threshold	Lets you specify the brake acceleration threshold in m/s^2 .
Curvature used for reference velocity calculation	<p>Lets you select which curvature is used for calculating the reference velocity:</p> <ul style="list-style-type: none"> ▪ Vehicle Trajectory: The vehicle trajectory uses a curve fitting of the vehicle trajectory to calculate the reference velocity. It is recommended to use this option. ▪ Road Network: The curvature of the road network is used. It is not recommended to use this option. It exists only for migration purposes.

Lateral Controller Types Lets you select the controller type. Refer to [Lateral Controller 1 \(ASM Environment Reference !\[\]\(c507f772dba2b921f86777f01218e570_img.jpg\)](#)) and [Lateral Controller 2 \(ASM Environment Reference !\[\]\(a75296508989caaa77a08d26cfccd4e5_img.jpg\)](#)).

Traffic dialog setting

Lets you specify the global settings for the traffic.

Fellow User Signals Lets you edit a table containing the fellow user signals. You can add a description to each user signal.

To add a new user signal, click in the Name cell and specify a name.

To delete a user signal, open its context menu and select **Delete**.

To sort the user signals, open the context menu and select **Move Up** or **Move Up**.

Fellow Trigger Lets you edit a table containing the fellow trigger signals. You can add a description to each trigger.

To add a new trigger, click in the Name cell and specify a name.

To delete a trigger, open its context menu and select **Delete**.

To sort the triggers, open the context menu and select **Move Up** or **Move Up**.

Global dialog settings

The dialog has a context menu with the following buttons commands.

Move up Lets you move up the selected row.

Move down Lets you move down the selected row.

Reset Lets you set the property values in this dialog according to the property values of the **Model Configuration** dialog.

Delete Lets you delete the selected row.

Related topics**HowTos**

[How to Specify a Lane Change Maneuver.....](#) 50

References

[Environment Configuration Dialog.....](#) 92
[Fellow User Signal Properties.....](#) 127
[Maneuver User Signal Properties.....](#) 159
[Properties.....](#) 116

Scenario Settings Dialog

Access

The Scenario Settings dialog opens when you select the **Settings** command.

Purpose

To configure settings for the simulation model.

Dialog settings

Minimum vector size (Simulink) Lets you specify the minimum vector size for the download. Use it to avoid recompiles of the simulation model in the MATLAB accelerator mode after downloading a modified scenario.

Related topics**HowTos**

[How to Create a Fellow User Signal.....](#) 60

References

[Settings.....](#) 116

Traffic Driver Dialog

Access	The Traffic Driver dialog opens when you select the New or Open command.
Purpose	To configure the properties of a traffic driver.
General page	<p>Author Lets you specify the name of the author.</p> <p>Description Lets you specify a description of the traffic driver.</p>
Driving Style page	<p>Distance at standstill Lets you specify the distance of the driver's vehicle to the vehicle ahead at standstill in meters.</p> <p>Distance Lets you specify the distance of the driver's vehicle to the vehicle ahead in seconds while driving. This value is used to calculate the required distance in meters.</p> <p>Detection range Lets you specify the range in front of the driver's vehicle in meters within other traffic participants are detected along the road reference line.</p> <p>Maximum longitudinal acceleration Lets you specify the maximum acceleration in longitudinal direction in everyday traffic in m/s^2.</p> <p>Comfortable deceleration Lets you specify the comfortable deceleration in everyday traffic in m/s^2.</p> <p>Maximum lateral acceleration Lets you specify the maximum lateral acceleration in curves in m/s^2.</p>
Rule-Compliant Driving page	<p>Distance at standstill Lets you specify the distance between the driver's vehicle and a traffic light or priority sign at standstill in meters.</p> <p>Speed limit offset Lets you specify the offset to the specified speed limit at which the driver's vehicle can drive slower or faster than the specified speed limit in km/h.</p> <p>Sign-compliant driving start Lets you specify the offset from the position of the speed limit or no passing start sign to the position where the driver's vehicle complies with the traffic sign in meters.</p> <p>Sign-compliant driving end Lets you specify the offset from the position of the speed limit or no passing end sign to the position where the driver's vehicle complies with the traffic sign in meters.</p> <p>Yield speed factor Lets you specify the deceleration factor before junctions in percent. This value is used to calculate the velocity the vehicle slows down when approaching a junction where the right-of-way status must be checked.</p>

Speed limit mode Lets you select the mode for road scenery speed limit use.

Value	Description
Ignore scenery	The speed limit specified by the road scenery is not evaluated.
Use scenery	The speed limit specified by the road scenery is evaluated.

Traffic Sign Recognition page

Distance front Lets you specify the distance for traffic sign recognition in front of the vehicle in meters.

Distance right Lets you specify the distance for traffic sign recognition to the right of the vehicle in meters.

Distance left Lets you specify the distance for traffic sign recognition to the left of the vehicle in meters.

Distance vertical Lets you specify the distance for traffic sign recognition above the vehicle in meters.

Relative sign angle Lets you specify the maximum angle between the traffic and the vector from the sign to the vehicle main point in degrees.

Related topics**HowTos**

[How to Specify the Properties of a Traffic Driver.....](#) 69

References

[New \(Traffic Driver\).....](#) 86
[Open \(Traffic Driver\).....](#) 87

Scenario Editor Creation Commands

Introduction

The Scenario Editor provides various commands to specify scenarios. They are accessible via the ribbon and context menus.

Where to go from here

Information in this section

Append Segment.....	104
To add a new segment after the last segment of the sequence.	
Append Sequence.....	105
To add a new sequence to the selected element.	
Collapse.....	105
To collapse the selected block.	
Collapse All.....	106
To collapse all the blocks.	
Copy.....	107
To copy an element to the Clipboard.	
Cut.....	107
To cut an element and copy it to the Clipboard.	
Delete <Element>.....	108
To delete an element.	
Delete Sequence.....	108
To delete a sequence.	
Expand.....	109
To expand a collapsed block.	
Expand All.....	110
To expand all the blocks.	
Insert Segment - After.....	110
To add a new segment after the selected segment.	
Insert Segment - Before.....	111
To add a new segment before the current segment.	
Layout Diagram.....	111
To arrange all the maneuver, fellow, and global user signal blocks on the Scenario pane.	
New – Fellow.....	112
To add a new fellow to the scenario.	
New – Maneuver.....	113
To add a maneuver to the scenario.	

New – User Signal.....	114
To add a new global user signal to the scenario.	
Paste.....	115
To paste an element from the Clipboard to the current position.	
Paste Sequence.....	115
To paste a sequence from the Clipboard to the current position.	
Properties.....	116
To configure the properties of the open scenario.	
Settings.....	116
To specify settings of the Scenario Editor.	

Append Segment

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – Segment
Shortcut key	None
Icon	None

Purpose

To add a new segment after the last segment of the sequence.

Result

A new segment is added to the sequence.

Related topics

HowTos

How to Add Sequences and Segments for the Maneuver.....	43
How to Initialize and Specify the Movements of a Fellow.....	63
How to Specify a Global User Signal.....	67

Append Sequence

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – block
Shortcut key	None
Icon	None

Purpose

To add a new sequence to the selected element.

Result

A new sequence is added to the maneuver, fellow, or global user signal.

Description

When you add a new sequence, it is created with two segments. The first segment should be used to initialize the element, especially the maneuver or a fellow. To specify the elements, select them and specify the properties in the **Properties** pane.

You can add further segments by using the **Insert Segment - Before** and **Insert Segment - After** commands.

Related topics

HowTos

How to Add Sequences and Segments for the Maneuver.....	43
How to Initialize and Specify the Movements of a Fellow.....	63

References

Delete Sequence.....	108
Insert Segment - After.....	110
Insert Segment - Before.....	111

Collapse

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – block – Change View
Shortcut key	None

Icon	None
Others	Double-click on an expanded block

Purpose To collapse the selected block.

Result The block collapsed.

Related topics


References

Collapse All.....	106
Expand.....	109
Expand All.....	110

Collapse All

Access

You can access this command via:

Ribbon	None
Context menu of	None
Shortcut key	None
Icon	 on the Scenario pane

Purpose To collapse all the blocks.

Result The blocks are collapsed.

Related topics

Basics

Filtering, Collapsing, or Expanding Blocks in the Scenario Editor.....	24
--	----

References

Collapse.....	105
Expand.....	109
Expand All.....	110

Copy

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – Maneuver Scenario pane – Fellow Scenario pane – Segment Scenario pane – Activity Scenario pane – Transition
Shortcut key	None
Icon	None

Purpose

To copy an element to the Clipboard.

Result

The selected element is copied to the Clipboard. You can paste it to another position.

Related topics

References

[Paste..... 115](#)

Cut

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – Maneuver Scenario pane – Fellow Scenario pane – Segment
Shortcut key	None
Icon	None

Purpose

To an selected element and copy it to the Clipboard.

Result The selected element is deleted from the current position and copied to the Clipboard. You can paste it to a new position.

Related topics**References**

[Paste..... 115](#)

Delete <Element>

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – Maneuver Scenario pane – Fellow Scenario pane – Global user signal Scenario pane – Segment
Shortcut key	None
Icon	None

Purpose

To delete an element.

Result

The selected element is deleted.

Related topics**References**

[Cut..... 107](#)

Delete Sequence

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – Sequence

Shortcut key	None
Icon	None

Purpose To delete a sequence.

Result The selected sequence is deleted.

Related topics

Basics

Scenarios.....	26
Workflow for Maneuver Creation.....	40
Workflow for Traffic Creation.....	59

Expand

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – block – Change View
Shortcut key	None
Icon	None
Others	Double-click on a collapsed block

Purpose To expand a collapsed block.

Result The block is expanded.

Related topics


References

Collapse.....	105
Collapse All.....	106
Expand All.....	110

Expand All

Access

You can access this command via:

Ribbon	None
Context menu of	None
Shortcut key	None
Icon	 on the Scenario pane

Purpose

To expand all the blocks.

Result

The blocks are expanded.

Related topics

Basics

[Filtering, Collapsing, or Expanding Blocks in the Scenario Editor..... 24](#)

References

[Collapse..... 105](#)
[Collapse All..... 106](#)
[Expand..... 109](#)

Insert Segment - After

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – Segment
Shortcut key	None
Icon	None

Purpose

To add a new segment after the selected segment.

Result

A new segment is added to the sequence.

Related topics**HowTos**

How to Add Sequences and Segments for the Maneuver.....	43
How to Initialize and Specify the Movements of a Fellow.....	63
How to Specify a Global User Signal.....	67

Insert Segment - Before

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – segment
Shortcut key	None
Icon	None

Purpose

To add a new segment before the current segment.

Result

A new segment is added to the sequence.

Related topics**HowTos**

How to Add Sequences and Segments for the Maneuver.....	43
How to Initialize and Specify the Movements of a Fellow.....	63
How to Specify a Global User Signal.....	67

Layout Diagram

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane
Shortcut key	None
Icon	None

Purpose	To arrange all the maneuver, fellow, and global user signal blocks on the Scenario pane.
Result	ModelDesk arranges all the blocks of the active Scenario pane and resets the zoom level.
Related topics	<p>Basics</p> <p>Filtering, Collapsing, or Expanding Blocks in the Scenario Editor..... 24</p> <p>Graphical User Interface of the Scenario Editor..... 22</p>

New – Fellow

Access	<p>You can access this command via:</p> <table border="1"> <tr> <td>Ribbon</td><td>None</td></tr> <tr> <td>Context menu of</td><td>Scenario pane</td></tr> <tr> <td>Shortcut key</td><td>None</td></tr> <tr> <td>Icon</td><td>None</td></tr> </table>	Ribbon	None	Context menu of	Scenario pane	Shortcut key	None	Icon	None
Ribbon	None								
Context menu of	Scenario pane								
Shortcut key	None								
Icon	None								
Purpose	To add a new fellow to the scenario.								
Result	A new fellow is added to the scenario and can be specified.								
Description	<p>When you add a new fellow, it is created with two segments. The first segment should be used to initialize the fellow. To specify the segments, select them and specify their properties in the Properties pane.</p> <p>You can add further segments by using the Insert Segment - Before and Insert Segment - After commands.</p>								

Related topics**HowTos**

[How to Create and Specify a Fellow.....](#) 62

References

[Fellow Properties.....](#) 126
[Insert Segment - After.....](#) 110
[Insert Segment - Before.....](#) 111
[Start Properties \(Fellow\).....](#) 163

New – Maneuver

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane
Shortcut key	None
Icon	None

Purpose

To add a maneuver to the scenario.

Result

A maneuver is added to the scenario and can be specified.

Description

The command is disabled if the scenario already contains a maneuver. A scenario must contain only one maneuver.

When you add a maneuver, it is created with one segment. To specify the segment, select it and specify their properties in the **Properties** pane.

You can add further segments by using the **Insert Segment - Before** and **Insert Segment - After** commands.

Related topics**HowTos**

[How to Start Scenario Creation.....](#) 41

References

[Insert Segment - After.....](#) 110
[Insert Segment - Before.....](#) 111
[Start Properties \(Maneuver\).....](#) 164

New – User Signal

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane
Shortcut key	None
Icon	None

Purpose

To add a new global user signal to the scenario.

Result

A new global user signal is added to the scenario.

Description

When you add a new global user signal, it is created with one segment. To specify the user signal, select the segment and specify its properties in the Properties pane.

Related topics**HowTos**

[How to Specify a Global User Signal.....](#) 67

References

[Start Properties \(Global User Signal\).....](#) 164

Paste

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – Fellow Scenario pane – Segment Scenario pane – Activity Scenario pane – Transition
Shortcut key	None
Icon	None

Purpose

To paste an element from the Clipboard to the current position.

Result

The element is pasted to the current position.

Related topics

References

Copy.....	107
Cut.....	107

Paste Sequence

Access

You can access this command via:

Ribbon	None
Context menu of	Scenario pane – Sequence
Shortcut key	None
Icon	None

Purpose

To paste a sequence from the Clipboard to the current position.

Result The sequence is pasted to the current position.

Related topics


References

Copy.....	107
Cut.....	107

Properties

Access

You can access this command via:

Ribbon	Environment – Scenario
Context menu of	None
Shortcut key	None
Icon	

Purpose

To configure the properties of the open scenario.

Result

The Scenario Properties dialog opens.

Related topics


References

Scenario Properties Dialog.....	98
---------------------------------	----

Settings

Access

You can access this command via:

Ribbon	Environment – Scenario
Context menu of	None
Shortcut key	None
Icon	

Purpose	To specify settings of the Scenario Editor.
Description	The Scenario Settings Dialog dialog opens.
Related topics	<div> <p>HowTos</p> <p>How to Create a Fellow User Signal..... 60</p> </div> <div> <p>References</p> <p>Scenario Settings Dialog..... 100</p> </div>

Scenario Editor Properties

Overview

The Scenario Editor provides a Properties pane to specify the properties of the active scenario.

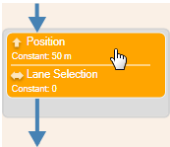
Where to go from here

Information in this section

Activity Properties (Fellow).....	119
To specify the properties of an activity of a fellow.	
Activity Properties (Global User Signal).....	120
To specify the properties of an activity of a global user signal.	
Activity Properties (Maneuver).....	121
To specify the properties of an activity of a maneuver.	
Clutch and Gear (MT) Properties.....	122
To specify the properties of the clutch and gear in an activity for an ASM vehicle with manual transmission.	
Fellow Properties.....	126
To specify the properties of a fellow.	
Fellow User Signal Properties.....	127
To specify a fellow user signal.	
Global User Signal Properties.....	129
To specify the properties of a global user signal.	
Lateral Type Properties (Fellow).....	129
To specify the properties of an activity in the lateral direction.	
Lateral Type Properties (Maneuver).....	133
To specify the properties of an activity in the lateral direction of the ASM vehicle.	
Longitudinal Type Properties (Fellow).....	143
To specify the properties of an activity in the longitudinal direction.	
Longitudinal Type Properties (Maneuver).....	151
To specify the properties of a maneuver's activity in the longitudinal direction.	
Maneuver Properties.....	158
To specify the properties of a maneuver.	
Maneuver User Signal Properties.....	159
To specify a maneuver user signal.	
Segment Properties.....	160
To specify the properties of a segment.	

Selector Lever (AT) Properties.....	161
To specify the properties of selector lever in an activity for an ASM vehicle with automatic transmission.	
Start Properties (Fellow).....	163
To specify the start properties of a fellow.	
Start Properties (Global User Signal).....	164
To specify the start properties of a global user signal.	
Start Properties (Maneuver).....	164
To specify the start properties of a maneuver.	
Traffic Scenario Properties.....	166
To specify the properties of a scenario.	
Transition Properties.....	167
To specify the properties of a transition.	
Source Properties.....	172
The source properties are used in several properties to specify the kind of source.	

Activity Properties (Fellow)

Purpose	To specify the properties of an activity of a fellow.
Access	To access the properties, click the activity on the Scenario pane. <div></div>
Properties	<div><div>Consistency</div><div>Displays the consistency state. If the specified properties are not consistent, the reasons for the inconsistency are given.</div></div> <div><div>Description</div><div>Lets you specify a description for the activity.</div></div> <div><div>Fellow User Signals</div><div>Lets you specify the fellow user signals. For details of the properties, refer to Fellow User Signal Properties on page 127</div></div> <div><div>Longitudinal</div><div>Lets you specify the activity in the longitudinal direction. For details of the properties, refer to Longitudinal Type Properties (Fellow) on page 143.</div></div>

Lateral Lets you specify the activity in the lateral direction. For details of the properties, refer to [Lateral Type Properties \(Fellow\)](#) on page 129.

Related topics

Basics

[Basics of Activities..... 31](#)

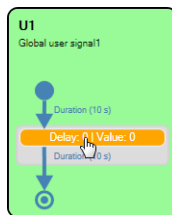
Activity Properties (Global User Signal)

Purpose

To specify the properties of an activity of a global user signal.

Access

To access the properties, click the activity on the **Traffic Scenario** pane.



Properties

Consistency Displays the consistency state. If the specified properties are not consistent, the reasons for the inconsistency are given.

Delay Lets you specify a time delay after which the specified value is applied after the start of the segment in seconds.

Description Lets you specify a description for the activity.

Parameters Lets you specify the parameters of the source type that is selected for the global user signal.

Type	Property	Description
Constant	Constant	Lets you specify a constant value.
Ramp	Start with current value	Lets you specify to start with the current value.
	Start	Lets you specify the start value.
	End	Lets you specify the end value.
	Extent	Lets you specify the time or distance that is required to change the user signal from the start value to the end value.
	Interpretation type	Lets you specify whether the value of Extent is specified in seconds or meters.

Type	Property	Description
Pulse	Value	Lets you specify the maximum value of the pulse.
	Step 1 start	Lets you specify the time or distance when the pulse starts.
	Step 1 end	Lets you specify the time or distance when the pulse reaches the specified value.
	Step 2 start	Lets you specify the time or distance when the starts falling.
	Step 2 end	Lets you specify the time or distance when the value is zero.
	Interpretation type	Lets you specify whether the start and end values are specified as time in seconds or distance in meters.
Table	Edit table	Lets you specify values in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify whether the values of the table are related to the time or distance.

Type Lets you specify the source type for the global user signal.

Related topics

Basics

[Basics of Activities.....](#) 31

References

[Edit Table Dialog.....](#) 91

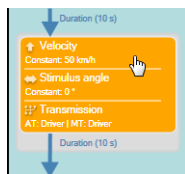
Activity Properties (Maneuver)

Purpose

To specify the properties of an activity of a maneuver.

Access

To access the properties, click the activity of the maneuver on the Scenario pane.



Properties

Clutch and gear (MT) Lets you specify the settings of the clutch and gear for an ASM vehicle that has a manual transmission. For details of the properties, refer to [Clutch and Gear \(MT\) Properties](#) on page 122.

Consistency Displays the consistency state. If the specified properties are inconsistent, the reasons for the inconsistency are given.

Description Lets you specify a description for the activity.

Longitudinal Lets you specify the activity in the longitudinal direction. For details of the properties, refer to [Longitudinal Type Properties \(Maneuver\)](#) on page 151.

Lateral Lets you specify the activity in the lateral direction. For details of the properties, refer to [Lateral Type Properties \(Maneuver\)](#) on page 133.

Maneuver User Signals Lets you select and specify maneuver user signal. For details of the properties, refer to [Maneuver User Signal Properties](#) on page 159.

Selector lever (AT) Lets you specify the selector lever for an ASM vehicle that has an automatic transmission. For details of the properties, refer to [Selector Lever \(AT\) Properties](#) on page 161.

Related topics**Basics**

[Basics of Activities.....](#) 31

HowTos

[How to Accelerate and Brake in a Maneuver.....](#) 45
[How to Follow a Road Automatically.....](#) 48
[How to Select and Specify Maneuver User Signals.....](#) 56
[How to Shift Gears in a Maneuver.....](#) 46
[How to Steer in a Maneuver.....](#) 52

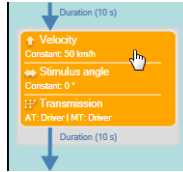
Clutch and Gear (MT) Properties

Purpose

To specify the properties of the clutch and gear in an activity for an ASM vehicle with manual transmission.

Access

To access the properties, click the activity of the maneuver on the Scenario pane.

**Properties**

Clutch and gear type Lets you specify the type of the clutch and gear settings of the maneuver.

Clutch and Gear Type	Description
Open clutch	Lets you specify an opened clutch. Refer to Open Clutch properties on page 123.
Driver	Lets you specify that the driver model specifies the clutch and gear. You can saturate the gear. Refer to Driver properties on page 123.
Stimulus	Lets you specify the clutch and gear settings are specified by properties in ModelDesk. Refer to Stimulus properties on page 123.
Reference gear	Lets you specify a specific gear used during the activity. Refer to Reference Gear properties on page 125.

Open Clutch properties

An Open Clutch type has no properties.

Driver properties

Saturate gear Lets you enable the saturation of the gear.

Gear Lets you enter a number specifying the highest gear the driver model can use.

Stimulus properties

Gear source type Lets you specify the gear source type.

Type	Description
Constant	Lets you specify a constant value for the gear. You can also use the gear used in the previous segment.
Extern	Lets you control the gear via Simulink model.
Manual	Lets you control the gear manually via experiment software.
Table	Lets you enter values for gears shift over time [s] or distance [m].

Gear source properties Lets you specify the properties of the selected gear source type.

Type	Property	Description
Constant	Hold current value	Lets you specify to use the gear of the previous segment.
	Gear	Lets you specify a specific gear.
Extern	–	Parameters are not required.
Manual	–	Parameters are not required.
Table	Edit table	Lets you edit a table that contains the values for stimulation. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify whether the values of the table are related to the time or distance.

Clutch source type Lets you specify the clutch source type.

Type	Description
Constant	Lets you specify a constant value for the clutch position or you can use the clutch position of the previous segment.
Ramp	Lets you specify a linear change of the clutch position.
Extern	Lets you control the clutch position via Simulink model.
Manual	Lets you control the clutch position manually via experiment software.
Table	Lets you enter values for clutch position over time [s] or distance [m].

Clutch source properties Lets you specify the properties of the selected clutch source type.

Type	Property	Description
Constant	Hold current value	Lets you specify to use the clutch position of the previous segment.
	Position	Lets you specify a specific clutch position.
Ramp	Start with current value	Lets you specify to start with the clutch position of the previous segment.
	Start value	Lets you specify the start value of the clutch position.
	End value	Lets you specify the end value of the clutch position.
	Extent	Lets you specify the time or distance that is required to change the clutch position from the start value to the end value.
	Interpretation type	Lets you specify whether the value of Extent is specified as time or distance.
Extern	–	Parameters are not required.

Type	Property	Description
Manual	–	Parameters are not required.
Table	Edit table	Lets you edit a table that contains the values for the position in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify whether the values of the table are related to the time or distance.

Reference Gear properties

Gear source type Lets you specify the gear source type.

Type	Description
Constant	Lets you enter a number to indicate one gear, for example, the number 1 for the first gear.
Extern	Lets you control the gear via Simulink model.
Manual	Lets you control the gear manually via experiment software.
Table	Lets you enter values for gears shift over time [s] or distance [m].

Gear source properties Lets you specify the properties of the selected gear source type.

Type	Property	Description
Constant	Hold current value	Lets you specify to use the gear of the previous segment.
	Gear	Lets you specify a specific gear.
Extern	–	Parameters are not required.
Manual	–	Parameters are not required.
Table	Edit table	Lets you edit a table that contains the values for the gear in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify whether the values of the table are related to the time or distance.

Related topics

HowTos

[How to Shift Gears in a Maneuver](#)..... 46

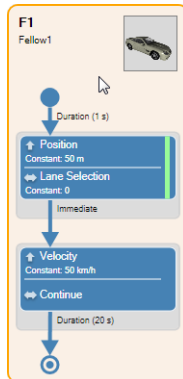
References

[Edit Table Dialog](#)..... 91

Fellow Properties

Purpose To specify the properties of a fellow.

Access To access the properties, click the fellow on the Scenario pane.



Properties

Consistency Displays the consistency state. If the specified properties are not consistent, the reasons for the inconsistency are given.

Description Lets you specify a description for the fellow.

Image Displays the selected traffic object.

Name Lets you specify the fellow's name.

Open Traffic Driver Opens the Traffic Driver dialog for you to edit the traffic driver. Refer to [Traffic Driver Dialog](#) on page 101.

Relative path Displays the relative path to the file of the traffic object.

Simulation ID Lets you specify the fellow's index. The index sets the order of the fellows in the traffic vector of the simulation model. Refer to [Traffic Scheduler \(ASM Traffic Reference\)](#).

Traffic Driver Lets you select the traffic driver that is used for the fellow. You can edit the object in a dialog, refer to [How to Specify the Properties of a Traffic Driver](#) on page 69.

Traffic Object Lets you select a traffic object to be used as the geometry for the fellow. When you click the button, the Traffic Object Browser opens to let you select a traffic object.

Tip

If the available traffic objects are not suitable for your fellow, you can create further traffic objects. Refer to [How to Create a Traffic Object \(ModelDesk Traffic Object Management\)](#).

Related topics**Basics**

[Basics of Fellows.....](#) 29

HowTos

[How to Create and Specify a Fellow.....](#) 62

References

[Manage Traffic Objects \(ModelDesk Traffic Object Management !\[\]\(642aa997563f9a325b310230bb5078b7_img.jpg\)\)](#)

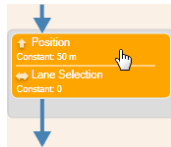
Fellow User Signal Properties

Purpose

To specify a fellow user signal.

Access

To access the properties, click the activity on the Scenario pane.

**Properties**

Delay Lets you specify a delay for the fellow user signal in seconds.

Index Displays the index of the selected fellow user signal.

Name Displays the name of the selected fellow user signal.

Parameters Lets you specify the parameters of the source type that is selected for the fellow user signal.

Type	Property	Description
Constant	Constant	Lets you specify a constant value.
Ramp	Start	Lets you specify the start value.
	End	Lets you specify the end value.
	Extent	Lets you specify the time or distance that is required to change the user signal from the start value to the end value.
	Interpretation type	Lets you specify whether the value of Extent is specified in seconds or meters.

Type	Property	Description
Pulse	Value	Lets you specify the maximum value of the pulse.
	Step 1 start	Lets you specify the time or distance when the pulse starts.
	Step 1 end	Lets you specify the time or distance when the pulse reaches the specified value.
	Step 2 start	Lets you specify the time or distance when the starts falling.
	Step 2 end	Lets you specify the time or distance when the value is zero.
	Interpretation type	Lets you specify whether the start and end values are specified as time in seconds or distance in meters.
Table	Edit table	Lets you specify values in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify whether the values of the table are related to the time or distance.

Type Lets you select the source type of the fellow user signal.

Valid Displays whether the specification of the fellow user signal is valid.

Related topics

Basics

[Basics of Fellows.....](#) 29

HowTos

[How to Create a Fellow User Signal.....](#) 60

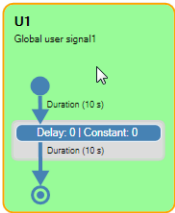
References

[Edit Table Dialog.....](#) 91

Global User Signal Properties

Purpose To specify the properties of a global user signal.

Access To access the properties, click the fellow on the Scenario pane.



Properties

Description Lets you specify a description for the global user signal.

Model Configuration Name Displays the name of the global user signal specified in the Model Configuration dialog. Refer to [Environment Configuration Dialog](#) on page 92.

Name Lets you specify the name of the global user signal.

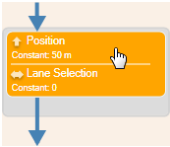
Simulation ID Lets you specify the global user signal's index.

Related topics	Basics
	Basics of Global User Signals..... 31
	HowTos
	How to Specify a Global User Signal..... 67

Lateral Type Properties (Fellow)

Purpose To specify the properties of an activity in the lateral direction.

Access To access the properties, click the activity on the Scenario pane.



Properties

Lateral type Lets you specify the longitudinal movement of a fellow in the activity.

Lateral Type	Description
Not used	Lets you specify that the fellow is not used in the specified activity. Refer to Not used properties on page 130.
Continue	Lets you specify that the fellow continues with the settings of the previous activity. Refer to Continue properties on page 130.
Lateral deviation	Lets you specify the lateral deviation of the fellow on the road. Refer to Lateral deviation properties on page 130.
Lane selection	Lets you specify the lane selection of a fellow. Refer to Lane selection properties on page 131.

Not used properties

A Not Used lateral type has no properties.

Continue properties

A Continue lateral type has no properties.

Lateral deviation properties

Reference type Lets you select whether the fellow's velocity is specified with absolute values or relative to a vehicle.

Reference vehicle Lets you select the reference vehicle when the Reference Type property is set to Relative.

Type Lets you select the profile type for the fellow's lateral deviation profile. The following types are possible:

- Constant: Stimulus with constant values
- Extern: Stimulus with a value defined by an external signal
- Final: Stimulus with a final value at the end of the current segment
- Ramp: Stimulus with a ramp profile
- Smooth lane change: Stimulus with a smooth sine profile
- Table: Stimulus with values of a table

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Hold current value	Lets you select whether the fellow's lateral deviation is kept from the end of the last segment until the end of the active segment.
	Constant	Lets you specify the fellow's lateral deviation.

Type	Property	Description
Extern	–	It is not necessary to specify a property for a stimulus with a value defined by an external signal.
Final	Final deviation	Lets you specify the target lateral deviation from the reference line . It can happen that the final deviation is not reached at the end of the segment. This depends on the lateral velocity and the duration of the segment).
	Lateral velocity	Lets you specify the fellow's maximum lateral velocity for the lateral deviation.
Ramp	Start deviation	Lets you specify the ramp profile's initial value for the fellow's lateral deviation.
	End deviation	Lets you specify the ramp profile's final value for the fellow's lateral deviation.
	Extent	Lets you specify the time for the fellow to change from the initial lateral deviation to the final lateral deviation.
Smooth lane change	Start deviation	Lets you specify the profile's initial value for the fellow's lateral deviation.
	End deviation	Lets you specify the profile's final value for the fellow's lateral deviation.
	Duration	Lets you specify the time for the fellow to change from the initial lateral deviation to the final lateral deviation.
Table	Edit table	Lets you specify values for the lateral deviation in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Lane selection properties

Force to road Lets you enable/disable that the nearest available lane index is used if the defined lane index is not available in the current road lane section.

Reference Type Lets you select whether the fellow's lane selection is specified with absolute values or relative to a vehicle.

Reference vehicle Lets you select the reference vehicle when the **Reference Type** property is **Relative**.

Type Lets you select the type for the fellow's lane selection. The following types are possible:

- **Constant:** The fellow constantly uses the same lane (either a specified lane or the lane of the previous segment).
- **Lane change:** The fellow changes lanes linearly within a specified time.
- **Smooth lane change:** The fellow changes lanes smoothly within a specified time.
- **Final:** Lane change from the lane used in the previous segment to a specified final lane. You can specify the lateral velocity.
- **Extern:** Lane selection with a value defined by an external signal. Non-integer values are allowed. For their interpretation, see [Road \(ASM Environment Reference\)](#).
- **Table:** Stimulus with values of a table.

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Hold current value	Lets you select whether the lane index is held from the end of the last segment until the end of the active segment.
	Lane index	Lets you specify the lane index used by the fellow.
Lane change	Start lane index	Lets you specify the index of the lane used by the fellow at the start of the segment.
	End lane index	Lets you specify the index of the lane used by the fellow at the end of the segment.
	Extent	Lets you specify the time for the fellow to change from the initial lane to the final lane.
Smooth lane change	Start lane index	Lets you specify the index of the lane used by the fellow at the start of the segment.
	End lane index	Lets you specify the index of the lane used by the fellow at the end of the segment.
	Extent	Lets you specify the time for the fellow to change from the initial lane to the final lane.
Final	Lateral velocity	Lets you specify the fellow's maximum lateral velocity necessary to reach the

Type	Property	Description
	Final lane index	specified lane at the end of the current segment. Lets you specify the index of the lane used by the fellow at the end of the current segment.
Extern	–	It is not necessary to specify a property for a stimulus with a value defined by an external signal.
Table	Edit table Interpretation type	Lets you specify values for the lane selection in relation to the time or distance. Refer to Edit Table Dialog on page 91. Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Related topics

HowTos

How to Create and Specify a Fellow.....	62
How to Specify a Global User Signal.....	67

References

Edit Table Dialog.....	91
--	--------------------

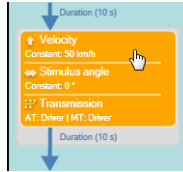
Lateral Type Properties (Maneuver)

Purpose

To specify the properties of an activity in the lateral direction of the ASM vehicle.

Access

To access the properties, click the activity of the maneuver on the Scenario pane.

**Properties**

Lateral type Lets you specify the movement of the ASM vehicle in lateral direction in the activity.

Lateral Type	Description
Basic road	Lets you specify that the ASM vehicle drives straight ahead or circular. Refer to Basic road properties on page 139.
Follow road	Lets you specify that the ASM vehicle follows the road. Refer to Follow road properties on page 139.
Not used	Lets you specify that the lateral movement is not changed in the activity. Refer to Not used properties on page 134.
Stimulus angle	Lets you specify that lateral movement is stimulated by specifying the steering angle. Refer to Stimulus angle properties on page 134.
Stimulus torque	Lets you specify that the lateral movement is stimulated by specifying the torque at the steering wheel. Refer to Stimulus torque properties on page 138.

Not used properties

A Not used lateral type has no properties.

Stimulus angle properties

Type Lets you specify the stimulus type.

Type	Description
Constant	Lets you specify a constant value for the steering angle. You can also continue with the steering angle of the previous segment.
Ramp	Lets you specify a linear change of the steering angle.
Extern	Lets you control the steering angle via Simulink model.
Manual	Lets you control the steering angle manually via experiment software.
Fixed	Lets you specify a fixed steering angle.
Step	Lets you specify stepwise change of the steering angle.
Pulse	Lets you specify a pulsed change of the steering angle.

Type	Description
Sine with dwell	Lets you specify a sine with dwell change of the steering angle.
Table	Lets you enter values for the steering angle over time or distance.

Parameters Lets you specify the parameters of the source type that is selected for the angle stimulus.

Type	Property	Description
Constant	Hold current value	Lets you select whether the steering angle is kept from the end of the previous segment.
	Angle	Lets you specify a constant steering angle in degrees.
Ramp	Start with current value	Lets you specify to start with the steering angle of the previous segment.
	Start angle	Lets you specify the start value of the steering angle in degrees.
	End angle	Lets you specify the end value of the steering angle in degrees.
	Extent	Lets you specify the duration or distance that is required to change the steering angle from the start value to the end value.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Extern	–	Parameters are not required.

Type	Property	Description
Sine	Start amplitude	Lets you specify the amplitude of the steering angle at the start in degrees.
	End amplitude	Lets you specify the amplitude of the steering angle at the end in degrees.
	Start frequency	Lets you specify the frequency of the steering angle at the start in Hz.
	End frequency	Lets you specify the frequency of the steering angle at the end in Hz.
	Offset	Lets you specify an offset value in degrees.
	Duration	Lets you specify the duration in seconds.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Manual	–	Parameters are not required.
Fixed	–	Parameters are not required.
Step	Direction	Lets you specify the direction.
	Angle	Lets you specify the steering angle in degrees.
	Step start	Lets you specify the start value.
	Step end	Lets you specify the end value.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Type	Property	Description
Pulse	Direction	Lets you specify the direction.
	Angle	Lets you specify the maximum steering angle.
	Step 1 start	Lets you specify the start value.
	Step 1 end	Lets you specify the end value (when the steering angle is the maximum angle).
	Step 2 start	Lets you specify the start value.
	Step 2 end	Lets you specify the end value (when the steering value is zero).
	Interpretation type	<p>Lets you specify how the property values are interpreted:</p> <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Sine with dwell	Direction	Lets you specify the direction.
	Angle	Lets you specify the amplitude of the steering angle in degrees.
	Frequency	Lets you specify the frequency in Hz.
	Dwell	Lets you specify the length of the dwell.
	Interpretation type	<p>Lets you specify how the property values are interpreted:</p> <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Table	Edit table	Lets you specify values for the steering angle in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	<p>Lets you specify how the property values are interpreted:</p> <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds.

Type	Property	Description
		<ul style="list-style-type: none"> ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Stimulus torque properties**Type** Lets you specify the stimulus type.

Type	Description
Constant	Lets you specify a constant value for the torque. You can also continue with the torque of the previous segment.
Ramp	Lets you specify a linear change of the torque.
Extern	Lets you control the torque via Simulink model.
Manual	Lets you control the torque manually via experiment software.
Table	Lets you enter values for the torque over time or distance.

Parameters Lets you specify the parameters of the source type that is selected for the torque stimulus.

Type	Property	Description
Constant	Hold current value	Lets you select whether the torque is kept from the end of the previous segment.
	Angle	Lets you specify a constant torque in Nm.
Ramp	Start with current value	Lets you specify to start with the torque of the previous segment.
	Start angle	Lets you specify the start value of the torque in Nm.
	End angle	Lets you specify the end value of the torque in Nm.
	Extent	Lets you specify the time or distance that is required to change the torque from the start value to the end value.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Extern	–	Parameters are not required.

Type	Property	Description
Manual	–	Parameters are not required.
Table	Edit table	Lets you specify values for the torque in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Basic road properties

Steering type Lets you specify the steering type.

Type	Description
Circle	Lets you specify that the ASM vehicle drives a circle with a constant radius.
Straight	Lets you specify that the ASM vehicle drives straight ahead.

Parameters Lets you specify the parameters of the source type that is selected for the steering type.

Type	Property	Description
Circle	Radius	Lets you specify the radius of the circle to drive in meters.
	Direction	Lets you specify the direction.
	Control mode	Lets you specify the control mode: <ul style="list-style-type: none"> ▪ Position: This is suitable for all maneuvers without road (i.e., with a basic road). ▪ Yaw rate: This control is without preview and is intended only for the circle lateral type, in particular for steady-state maneuvers. For more information, refer to Lateral Controller 2 (ASM Environment Reference) .
Straight	–	Parameters are not required.

Follow road properties

Adapt velocity to road Lets you specify that the driver adapts the velocity to the road.

Driver type Lets you select the driver type. You can use a predefined driver type (refer to [Scenario Properties Dialog](#) on page 98) or specify a user-defined driver.

Maximum longitudinal acceleration Lets you specify the maximum acceleration in longitudinal direction in m/s^2 .

Maximum lateral acceleration Lets you specify the maximum acceleration in lateral direction in m/s^2 .

Brake acceleration threshold Lets you specify the brake acceleration threshold in m/s^2 .

Enable lateral driver tolerance Lets you enable the lateral driver tolerance.

Lateral driver tolerance Lets you specify the lateral driver tolerance in meters. For more information, refer to [Lateral Controller 1 \(ASM Environment Reference !\[\]\(23d9fc146e83b5c3013cfa32c784f8d5_img.jpg\)](#)) and [Lateral Controller 2 \(ASM Environment Reference !\[\]\(f5c463b8c1554ac5049d611bd8e33a51_img.jpg\)](#).

Reference type Lets you specify how the lane index is interpreted. For details, refer to [How to Specify a Lane Change Maneuver](#) on page 50.

Reference Type	Description
Relative to lane	The lane index is interpreted as specified in the global properties in the scenario settings. Refer to Scenario Properties Dialog on page 98.
Relative to object	The lane index is interpreted relative to the lane index of another reference object. The route and the direction of travel of the reference object and the fellow vehicle must be identical. Lanes to the right of the fellow vehicle in relation to its direction of travel have negative indices. Lane to the left have positive indices.

Reference object Displays the reference object to which the lane index is related. It is limited to the ASM vehicle.

Lane index type Lets you specify the lane index type.

Lane Index Type	Description
Constant	Lets you specify a lane index where the ASM vehicle drives or specify that the ASM vehicle stays on the current lane.
Extern	Lets you specify the lane index via Simulink model.
Final	Lets you specify a lane change by specifying the target lane and a lateral velocity.
Manual	Lets you specify the lane index via experiment software.
Table	Lets you enter values for the lane index over time or distance.

Parameters (lane index) Lets you specify the parameters of the source type that is used as lane index type. Non-integer values are allowed as lane index. For their interpretation, refer to [Road \(ASM Environment Reference !\[\]\(758ebdf4629c903da74c2e079717ae32_img.jpg\)](#).

Type	Property	Description
Constant	Hold current value	Lets you select that the ASM vehicle stays on the current lane.
	Lane index	Lets you specify the lane where the ASM vehicle drives.
Extern	–	Parameters are not required.
Final	Final lane index	Lets you specify the target lane.
	Lateral velocity	Lets you specify the lateral velocity of the ASM vehicle when it changes the lane.
Manual	–	Parameters are not required.
Table	Edit table	Lets you specify values for the lane index in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	<p>Lets you specify how the property values are interpreted:</p> <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Force to road Lets you select whether the ASM vehicle is forced to stay on the road. If the option is set and a lane outside the road is selected, the ASM vehicle remains on the road and drives on the outermost available lane. Otherwise, the vehicle drives besides the road.

Lateral offset type Lets you specify the lateral offset type.

Lane Index Type	Description
Constant	Lets you specify a constant distance of the ASM vehicle to the lane in meters. You can specify a specific value or use the value of the previous segment.
Ramp	Lets you specify a linear change of the distance of the ASM vehicle to the lane.
Extern	Lets you specify the distance of the ASM vehicle to the lane via Simulink model.
Manual	Lets you specify the distance of the ASM vehicle to the lane via experiment software.

Lane Index Type	Description
Table	Lets you enter values for the distance of the ASM vehicle to the lane over time or distance.

Parameters (lateral offset) Lets you specify the parameters of the source type that is used as lateral offset type.


Type	Property	Description
Constant	Hold current value	Lets you select that the ASM vehicle stays on the current lane.
	Lateral offset	Lets you specify the lane where the ASM vehicle drives.
Ramp	Start with current value	Lets you specify to use the distance of the ASM vehicle to the lane of the previous lane.
	Start offset	Lets you specify the distance of the ASM vehicle to the lane at the start.
	End offset	Lets you specify the distance of the ASM vehicle to the lane at the end.
	Extent	Lets you specify the time or distance that is required to change the offset from the start value to the end value.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Extern	–	Parameters are not required.
Manual	–	Parameters are not required.
Table	Edit table	Lets you specify values for the lane index in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Related topics

HowTos

How to Follow a Road Automatically.....	48
How to Specify a Lane Change Maneuver.....	50
How to Steer in a Maneuver.....	52

References

Edit Table Dialog.....	91
Lateral Controller 1 (ASM Environment Reference )	
Lateral Controller 2 (ASM Environment Reference )	

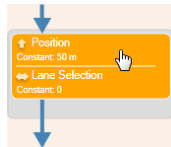
Longitudinal Type Properties (Fellow)

Purpose

To specify the properties of an activity in the longitudinal direction.

Access

To access the properties, click the activity on the Scenario pane.



Properties

Longitudinal type Lets you specify the longitudinal movement of a fellow in the activity.

Longitudinal Type	Description
Not used	Lets you specify that the fellow is not used in the specified activity. Refer to Not used properties on page 144.
Continue	Lets you specify that the fellow continues with the settings of the previous activity. Refer to Continue properties on page 144.
Acceleration	Lets you specify the acceleration of the fellow. Refer to Acceleration properties on page 144.
Distance [m]	Lets you specify the fellow's distance in meters relative to another fellow or the ASM vehicle. Refer to Distance m properties on page 145.
Distance [s]	Lets you specify the fellow's distance in seconds relative to another fellow or the ASM vehicle. Refer to Distance s properties on page 146.

Longitudinal Type	Description
Position	Lets you specify the longitudinal position of the fellow from the road start position. Refer to Position properties on page 147.
Traffic Driver	Lets you use a traffic driver for the fellow. Refer to Traffic Driver properties on page 148.
Velocity	Lets you specify the velocity of the fellow. Refer to Velocity properties on page 148.
Stop	Lets you specify the fellow's deceleration until stopping within a specified distance. Refer to Stop properties on page 150.

Not used properties A Not Used longitudinal type has no properties.

Continue properties A Continue longitudinal type has no properties.

Acceleration properties

Fellow dynamics Lets you select the behavior of the fellow for realistic behavior in curves and acceleration:

- Activated: The fellow behavior is specified with the properties of the selected traffic object.
- Deactivated: The fellow drives exactly with the specified acceleration.

Type Lets you select the profile type for the fellow's acceleration profile. The following types are possible:

- Constant: Stimulus with constant values
- Ramp: Stimulus with a ramp profile
- Table: Stimulus with values of a table

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Acceleration	Lets you specify the acceleration in m/s^2 .
Ramp	Start acceleration	Lets you specify the ramp profile's initial value for acceleration in the longitudinal direction in m/s^2 .
	Duration	Lets you specify for how long the fellow accelerates in the longitudinal direction in seconds.
	End acceleration	Lets you specify the ramp profile's final value for acceleration in the longitudinal direction in m/s^2 .

Type	Property	Description
Table	Edit table	Lets you specify values for the acceleration. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> Valid only Time (Segment): The values are a time duration in seconds.

Distance m properties

Reference point Lets you select the reference point of the selected fellow.

Reference object Lets you select the object, for example, a vehicle, relative to which the fellow's distance is specified.

Point on reference object Lets you select the reference point of the referenced object that is used in the calculation of the distance.

Type Lets you select the profile type for the fellow's distance profile. The following types are possible:

- Constant: Stimulus with constant values
- Ramp: Stimulus with a ramp profile
- Extern: Stimulus with a value defined by an external signal
- Final: Stimulus with a final value at the end of the current segment
- Table: Stimulus with values of a table

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Hold current value	Lets you select whether the fellow's distance in meters relative to another vehicle is held from the end of the last segment until the end of the active segment.
	Distance	Lets you specify the fellow's distance in meters relative to another vehicle.
Extern	–	It is not necessary to specify a property for a stimulus with a value defined by an external signal.
Final	Distance	Lets you specify the fellow's distance in meters relative to another vehicle.
	Relative velocity	Lets you specify the fellow's velocity relative to another vehicle at the end of the current segment.

Type	Property	Description
Ramp	Start distance	Lets you specify the ramp profile's initial value for the fellow's distance in meters relative to another vehicle.
	End distance	Lets you specify the ramp profile's final value for the fellow's distance in meters relative to another vehicle.
	Duration	Lets you specify the time for the fellow to move from the start distance to the end distance.
Table	Edit table	Lets you specify values for the distance in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Distance s properties

Reference vehicle Lets you select the vehicle relative to which the fellow's distance is specified.

Type Lets you select the profile type for the fellow's distance profile. The following types are possible:

- **Constant:** Stimulus with constant values
- **Ramp:** Stimulus with a ramp profile
- **Extern:** Stimulus with a value defined by an external signal
- **Final:** Stimulus with a final value at the end of the current segment
- **Table:** Stimulus with values of a table

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Hold current value	Lets you select whether the fellow's distance in seconds relative to another vehicle is held from the end of the last segment until the end of the active segment.
	Distance	Lets you specify the fellow's distance in seconds relative to another vehicle.

Type	Property	Description
Extern	–	It is not necessary to specify a property for a stimulus with a value defined by an external signal.
Final	Distance	Lets you specify the fellow's distance in seconds relative to another vehicle.
	Relative velocity	Lets you specify the fellow's velocity relative to another vehicle at the end of the current segment.
Ramp	Start distance	Lets you specify the ramp profile's initial value for the fellow's distance in seconds relative to another vehicle.
	End distance	Lets you specify the ramp profile's final value for the fellow's distance in seconds relative to another vehicle.
	Duration	Lets you specify the time for the fellow to move from the start distance to the end distance.
Table	Edit table	Lets you specify values for the distance in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Position properties

Type Lets you select the profile type for the fellow's position profile. The following types are possible:

- **Constant**: Stimulus with constant values
- **Ramp**: Stimulus with a ramp profile
- **Extern**: Stimulus with a value defined by an external signal
- **Table**: Stimulus with values of a table

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Hold current value	Lets you select whether the fellow's longitudinal position is held from the end of

Type	Property	Description
	Position	the last segment until the end of the active segment. Lets you specify the fellow's longitudinal position from the road start position.
	Yaw angle	Lets you specify the fellow's initial orientation relative to the direction of the road reference line.
Extern	–	It is not necessary to specify a property for a stimulus with a value defined by an external signal.
Ramp	Start position	Lets you specify the ramp profile's initial value for the fellow's longitudinal position.
	End position	Lets you specify the ramp profile's final value for the fellow's longitudinal position.
	Duration	Lets you specify the time for the fellow to move from the start position to the end position.
Table	Edit table	Lets you specify values for the distance in relation to the time. Refer to Edit Table Dialog on page 91.

Traffic Driver properties

Type Lets you select the profile type for the fellow's distance profile. The following types are possible:

- Constant: Stimulus with constant values

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Hold current value	Lets you select whether the fellow's velocity is held from the end of the last segment until the end of the active segment.
	Distance	Lets you specify the fellow's velocity in km/h.

Velocity properties

Fellow dynamics Lets you select the behavior of the fellow for realistic behavior in curves and acceleration:

- Activated: The fellow behavior is specified with the properties of the selected traffic object.
- Deactivated: The fellow drives exactly with the specified velocity.

Reference type Lets you select whether the fellow's velocity is specified with absolute values or relative to a vehicle.

Reference vehicle Lets you select the reference vehicle when the Reference Type property is set to Relative.

Type Lets you select the profile type for the fellow's velocity profile. The following types are possible:

- **Constant:** Stimulus with constant values
- **Ramp:** Stimulus with a ramp profile
- **Extern:** Stimulus with a value defined by an external signal
- **Final:** Stimulus with a final value at the end of the current segment
- **Sine:** Stimulus with a sinusoidal form in the current segment
- **Table:** Stimulus with values of a table

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Hold current value	Lets you select whether the fellow's velocity is held from the end of the last segment until the end of the active segment.
	Constant	Lets you specify the fellow's velocity.
Extern	–	Parameters are not required.
Final	Final velocity	Lets you specify the fellow's velocity at the end of the current segment.
	Positive acceleration	Lets you specify the acceleration value for reaching the final value.
	Negative acceleration	Lets you specify the deceleration value for reaching the final value.
Ramp	Start velocity	Lets you specify the ramp profile's initial value for the fellow's velocity.
	End velocity	Lets you specify the ramp profile's final value for the fellow's velocity.
	Extent	Lets you specify the time for the fellow to accelerate or decelerate from the start velocity to the end velocity.
Sine	Amplitude	Lets you specify the amplitude of the velocity in km/h.
	Start frequency	Lets you specify the initial frequency of the sine wave in Hz.
	End frequency	Lets you specify the final frequency of the sine wave in Hz.
	Offset	Lets you specify an offset value.
	Extent	Lets you specify the extent (duration or distance) for the fellow to accelerate and brake in a sinusoidal form from the start frequency to the end frequency.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds.

Type	Property	Description
		<ul style="list-style-type: none"> ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Table	Edit table Interpretation type	<p>Lets you specify values for the velocity in relation to the time or distance. Refer to Edit Table Dialog on page 91.</p> <p>Lets you specify how the property values are interpreted:</p> <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Stop properties

Type (read-only) Displays the constant profile as the acceleration stimulus type.

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Stop within distance	Hold current value	Lets you specify to use the velocity of the previous segment for the fellow. If you enable the option, you do not have to specify the Velocity property.
	Velocity	Lets you specify the velocity of the fellow at the beginning of the segment in km/h.
	Deceleration	Lets you specify a deceleration for the braking process in m/s^2 .
	Stop distance	Lets you specify the distance in meters in which the fellow must stop.

Related topics

HowTos

[How to Create and Specify a Fellow.....](#) 62

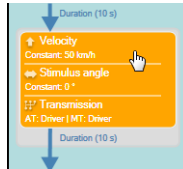
References

[Edit Table Dialog.....](#) 91

Longitudinal Type Properties (Maneuver)

Purpose To specify the properties of a maneuver's activity in the longitudinal direction.

Access To access the properties, click the activity of the maneuver on the **Scenario** pane.



Properties **Longitudinal type** Lets you specify the longitudinal movement of the ASM vehicle in the activity.

Longitudinal Type	Description
Lateral acceleration	Lets you specify the lateral acceleration of the ASM vehicle. All the longitudinal definitions of the current segment are ignored. The longitudinal velocity depends on the lateral acceleration. This type can be used only if the lateral type is Basic road and the steering type is Circle . Refer to Lateral acceleration properties on page 151.
Pedal stimulus	Lets you specify the stimulation of the acceleration pedal and brake pedal. Refer to Pedal stimulus properties on page 152.
Standstill	Lets you specify the standstill of the ASM vehicle. Refer to Standstill properties on page 155.
Stop	Lets you specify to stop the ASM vehicle within a specified distance or brake until stop. Refer to Stop properties on page 155.
Traffic Driver	Lets you use the selected traffic driver for the ASM vehicle. Refer to Traffic Driver properties on page 155.
Velocity	Lets you specify the velocity of the ASM vehicle. Refer to Velocity properties on page 156.

Lateral acceleration properties

Type Lets you select the profile type for the lateral acceleration. The following types are possible:

- **Constant:** Specifies the lateral acceleration using a constant values
- **Ramp:** Specifies the lateral acceleration using a ramp profile
- **Table:** Specifies the lateral acceleration using a table

Parameters Lets you specify the parameters of the source type that is selected for the lateral acceleration.

Type	Property	Description
Constant	Lateral acceleration	Lets you specify a constant value for the lateral acceleration in m/s^2 .
Ramp	Start acceleration End acceleration Extent Interpretation type	<p>Lets you specify the start value of the ramp profile in m/s^2.</p> <p>Lets you specify the end value of the ramp profile in m/s^2.</p> <p>Lets you specify the distance or duration that is used to change from the start value to the end value.</p> <p>Lets you specify how the property values are interpreted:</p> <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Table	Edit table Interpretation type	<p>Lets you specify values for lateral acceleration in relation to the time or distance. Refer to Edit Table Dialog on page 91.</p> <p>Lets you specify how the property values are interpreted:</p> <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Pedal stimulus properties

Acceleration pedal type Lets you select the profile type for the stimulation of the acceleration pedal. The following types are possible:

- **Constant:** Stimulates the pedal with constant values
- **Ramp:** Stimulates the pedal with a ramp profile
- **Extern:** Stimulates the pedal with a value defined by the simulation model
- **Manual:** Stimulates the pedal with a value defined by the experiment software
- **Table:** Stimulates the pedal with values defined in a table

Acceleration pedal parameters Lets you specify the parameters of the source type that is selected for the acceleration pedal.

Type	Property	Description
Constant	Hold current value	Lets you select to use the pedal position of the end of the previous segment.
	Position	Lets you specify a constant value for the pedal position in percents.
Ramp	Start with current value	Lets you specify to start the ramp profile with the pedal position of the end of the previous segment.
	Start value	Lets you specify the start value of the ramp profile in percents.
	End value	Lets you specify the end value of the ramp profile in percents.
	Extent	Lets you specify the distance or duration that is used to change from the start value to the end value.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Extern	–	Parameters are not required.
Manual	–	Parameters are not required.
Table	Edit table	Lets you specify values for pedal positions in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Brake pedal type Lets you select the profile type for the stimulation of the brake pedal. The following types are possible:

- **Constant:** Stimulates the pedal with constant values
- **Ramp:** Stimulates the pedal with a ramp profile
- **Extern:** Stimulates the pedal with a value defined by the simulation model
- **Manual:** Stimulates the pedal with a value defined by the experiment software
- **Table:** Stimulates the pedal with values defined in a table

Brake pedal parameters Lets you specify the parameters of the source type that is selected for the brake pedal.

Type	Property	Description
Constant	Hold current value	Lets you select to use the pedal position of the end of the previous segment.
	Position	Lets you specify a constant value for the pedal position in percents.
Ramp	Start with current value	Lets you specify to start the ramp profile with the pedal position of the end of the previous segment.
	Start value	Lets you specify the start value of the ramp profile in percents.
	End value	Lets you specify the end value of the ramp profile in percents.
	Extent	Lets you specify the distance or duration that is used to change from the start value to the end value.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Extern	–	Parameters are not required.
Manual	–	Parameters are not required.
Table	Edit table	Lets you specify values for pedal positions in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds.

Type	Property	Description
		<ul style="list-style-type: none"> ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Standstill properties

A Standstill longitudinal type has no properties.

Stop properties

Type Lets you select the type of stop. The following types are possible:

- **Stop within distance:** Stops the ASM vehicle within a specified distance.
- **Brake until stop:** Brakes the ASM vehicle until it stops.

Parameters Lets you specify the parameters of the source type that is selected for the distance m.

Type	Property	Description
Stop within distance	Hold current value	Lets you select whether the velocity of the previous segment is used at the beginning of the segment.
	Velocity	Lets you specify the velocity at the beginning of the segment.
	Deceleration	Lets you specify the deceleration of the ASM vehicle in m/s ² .
	Stop distance	Lets you specify the distance at which the ASM vehicle must stop in meters.
Brake until stop	–	Parameters are not required.

Traffic Driver properties

Type Lets you select the profile type for the fellow's distance profile. The following types are possible:

- **Constant:** Stimulus with constant values

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Hold current value	Lets you select whether the ASM vehicle's velocity is held from the end of the last segment until the end of the active segment.
	Distance	Lets you specify the ASM vehicle's velocity in km/h.

Velocity properties

Reference type Lets you specify whether the velocity is specified absolutely or relatively to a vehicle.

Reference vehicle Lets you select the vehicle if the velocity is relative to a vehicle.

Type Lets you select the type for the velocity. The following types are possible:

- **Constant:** Lets you specify a constant value as velocity
- **Ramp:** Lets you specify a ramp profile for the velocity
- **Extern:** Lets you specify the velocity via simulation model
- **Sine:** Lets you specify a sine value as velocity
- **Final:** Lets you specify a final value at the end of the current segment
- **Manual:** Lets you control the velocity manually via experiment software.
- **Table:** Lets you specify the velocity using values of a table

Parameters Lets you specify the parameters of the source type that is selected for the velocity.

Type	Property	Description
Constant	Hold current value	Lets you select whether the velocity of the previous segment is hold.
	Velocity	Lets you specify a constant velocity.
Ramp	Start with current value	Lets you specify to start the ramp using the velocity at the end of the previous segment.
	Start velocity	Lets you specify the start value for the velocity.
	End velocity	Lets you specify the end value of the velocity.
	Extent	Lets you specify the distance or duration that is used to change from the start value to the end value.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Extern	–	Parameters are not required.

Type	Property	Description
Sine	Start amplitude	Lets you specify the amplitude of the velocity at the start in km/h.
	End amplitude	Lets you specify the amplitude of the velocity at the end in km/h.
	Start frequency	Lets you specify the frequency of the velocity at the start in Hz.
	End frequency	Lets you specify the frequency of the velocity at the end in Hz.
	Offset	Lets you specify an offset to the velocity in km/h.
	Extent	Lets you specify the distance or duration that is used to change from the start value to the end value.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.
Final	Final velocity	Lets you specify the velocity at the end of the segment.
	Positive acceleration	Lets you specify the acceleration in m/s^2
	Negative acceleration	Lets you specify the deceleration in m/s^2
Manual	–	Parameters are not required.
Table	Edit table	Lets you specify values for the velocity in relation to the time or distance. Refer to Edit Table Dialog on page 91.
	Interpretation type	Lets you specify how the property values are interpreted: <ul style="list-style-type: none"> ▪ Time (Segment): The values are a time duration in seconds. ▪ Driven Distance (Segment): The values are the distance driven by the vehicle in meters. ▪ Distance (Route): The values are the distance in relation to the reference line of the road (s coordinate) in meters.

Related topics

HowTos

How to Accelerate and Brake in a Maneuver..... 45

References

Edit Table Dialog..... 91

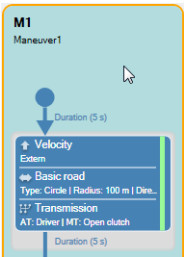
Maneuver Properties

Purpose

To specify the properties of a maneuver.

Access

To access the properties, click the maneuver on the Scenario pane.



Properties

Consistency Displays the consistency state. If the specified properties are not consistent, the reasons for the inconsistency are given.

Description Lets you specify a description for the maneuver.

Name Lets you specify the name of the maneuver.

Open Traffic Driver Opens the Traffic Driver dialog for you to specify a new traffic driver. Refer to [Traffic Driver Dialog](#) on page 101.

Relative Path Displays the relative path to the traffic object that is used as the contour line object.

Traffic Driver Lets you select the traffic driver that is used for the ASM vehicle in the scenario. You can edit the object in a dialog, refer to [How to Specify the Properties of a Traffic Driver](#) on page 69.

Vehicle Object Lets you select a traffic object whose contour line is used as the contour line for the ASM vehicle.

You can create traffic objects using the Traffic Object Manager. Refer to [Basics of Traffic Objects \(ModelDesk Traffic Object Management\)](#).

Related topics

Basics

[Basics of Maneuvers..... 27](#)

HowTos

[How to Start Scenario Creation..... 41](#)

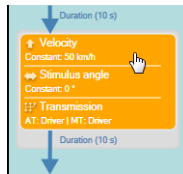
Maneuver User Signal Properties

Purpose

To specify a maneuver user signal.

Access

To access the properties, click the activity of the maneuver on the **Scenario** pane.



Properties

Delay Lets you specify a delay for the maneuver user signal in seconds.

Index Displays the index of the selected maneuver user signal.

Name Displays the name of the selected maneuver user signal.

Parameters Lets you specify the parameters of the source type that is selected for the maneuver user signal.

Type	Property	Description
Constant	Constant	Lets you specify a constant value.
Ramp	Start	Lets you specify the start value.
	End	Lets you specify the end value.
	Extent	Lets you specify the time that is required to change the user signal from the start value to the end value.
Pulse	Value	Lets you specify the maximum value of the pulse.
	Step 1 start	Lets you specify the time when the pulse starts.
	Step 1 end	Lets you specify the time when the pulse reaches the specified value.
	Step 2 start	Lets you specify the time when the starts falling.
	Step 2 end	Lets you specify the time when the value is zero.

Type	Property	Description
Table	Edit table	Lets you specify values in relation to the time. Refer to Edit Table Dialog on page 91.

Type Lets you select the source type of the maneuver user signal.

Valid Displays whether the specification of the maneuver user signal is valid.

Related topics

Basics

[Basics of Maneuvers..... 27](#)

HowTos

[How to Select and Specify Maneuver User Signals..... 56](#)

References

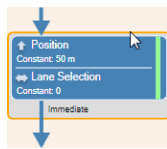
[Edit Table Dialog..... 91](#)

[Scenario Properties Dialog..... 98](#)

Segment Properties

Purpose To specify the properties of a segment.

Access To access the properties, click the segment on the Traffic Scenario pane.



Properties

Consistency Displays the consistency state. If the specified properties are not consistent, the reasons for the inconsistency are given.

Description Lets you enter a description for the segment.

Index Displays the index of the segment.

Name Lets you specify the name of the segment.

Related topics

HowTos

How to Create and Specify a Fellow.....	62
How to Specify a Global User Signal.....	67

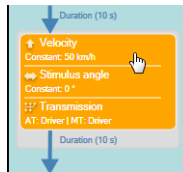
Selector Lever (AT) Properties

Purpose

To specify the properties of selector lever in an activity for an ASM vehicle with automatic transmission.

Access

To access the properties, click the activity of the maneuver on the Scenario pane.



Properties

Selector lever type Lets you specify the selector lever type of the ASM vehicle in the activity.

Selector Lever Type	Description
Driver	Lets you specify that the driver model specifies the selector lever. Refer to Driver properties on page 161.
Stimulus	Lets you specify the selector lever settings are specified by properties in ModelDesk. Refer to Stimulus properties on page 161.

Driver properties

A Driver selector lever type has no properties.

Stimulus properties

Type Lets you specify the stimulus type.

Type	Description
Constant	Lets you specify a constant value for the position. You can also use the position of the previous segment.
Ramp	Lets you specify a linear change of the position.
Extern	Lets you control the selector lever via Simulink model.

Type	Description
Manual	Lets you control the selector lever manually via experiment software.
Table	Lets you enter values for position shift over time [s] or distance [m].

Parameters Lets you specify the properties of the selected type.

Type	Property	Description
Constant	Hold current value	Lets you specify to use the position of the previous segment.
	Position	Lets you specify a specific position.
Ramp	Start with current value	Lets you specify to start with the position of the previous segment.
	Start value	Lets you specify the start value for the position.
	End value	Lets you specify the end value for the position.
	Extent	Lets you specify the time or distance that is required to change the position from the start value to the end value.
	Interpretation type	Lets you specify whether the value of Extent is related to the time or distance.
Extern	–	Parameters are not required.
Manual	–	Parameters are not required.
Type	Edit table	Lets you edit a table that contains the values for the position in relation to the time or distance.
	Interpretation type	Lets you specify whether the values of the table are related to the time or distance.

Related topics

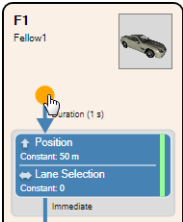
HowTos

[How to Shift Gears in a Maneuver.....](#) 46

Start Properties (Fellow)

Purpose To specify the start properties of a fellow.

Access To access the properties, click the start of a fellow on the Scenario pane.



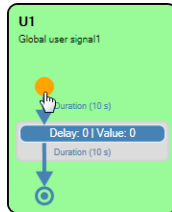
- Properties**
- Consistency** Displays the consistency state. If the specified properties are not consistent, the reasons for the inconsistency are given.
 - Description** Lets you specify a description for the start segment.
 - Name** Lets you specify the name of the start segment.
 - Reverse direction** Lets you specify that the fellow drives the route in the reverse direction. The route must be suitable for the reverse direction.
 - Reverse possible** Displays whether it is possible to drive the route in the reverse direction.
 - Route** Lets you select the route that the fellow drives. Usually, the entries display the indices and names of the routes. If only the index is displayed, the previously selected route is no longer available.
Routes are specified in the Road Generator. For more information, refer to [How to Specify Routes on Road Networks \(ModelDesk Road Creation\)](#).
 - Route index** Lets you select the route that the fellow drives.
The Route index specifies the same route as the Route property but you can create an alias for Route index. You can therefore use the alias feature with this property.
 - Use external** Lets you specify to use external route definitions.

Related topics	Basics
	Basics of Fellows..... 29

Start Properties (Global User Signal)

Purpose To specify the start properties of a global user signal.

Access To access the properties, click the start of a global user signal on the Traffic Scenario pane.



Properties

Consistency Displays the consistency state. If the specified properties are not consistent, the reasons for the inconsistency are given.

Description Lets you specify a description for the start segment.

Name Lets you specify the name of the start segment.

Related topics

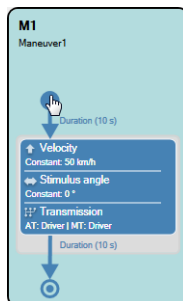
Basics

Basics of Global User Signals..... 31

Start Properties (Maneuver)

Purpose To specify the start properties of a maneuver.

Access To access the properties, click the start of a maneuver on the Scenario pane.



Start properties

Description Lets you specify a description for the start segment.

Name Lets you specify the name of the start segment.

Reverse direction Lets you specify that the fellow drives the route in the reverse direction. The route must be suitable for the reverse direction.

Reverse possible Displays whether it is possible to drive the route in the reverse direction.

Route Lets you select the route that the ASM vehicle drives. Usually, the entries display the indices and names of the routes. If only the index is displayed, the previously selected route is no longer available.

Routes are specified in the **Road Generator**. For details, refer to [How to Specify Routes on Road Networks \(ModelDesk Road Creation !\[\]\(830769b31eeeaca920791081939ff8ba_img.jpg\)](#)).

Route index Lets you select the route that the ASM vehicle drives. The Route index specifies the same route as the Route property but you can create an alias for Route index. You can therefore use the alias feature with this property.

Initial vehicle position

Additional lateral offset Lets you specify an additional lateral offset of the ASM vehicle position on the road in meters. This is a d coordinate in the coordinate system related to the selected lane.

Initial lane index Lets you specify the initial lane index.

Start position on route Lets you specify the start position of the ASM vehicle on the road in meters.

Vehicle orientation Lets you specify an initial relative yaw angle in degrees.

Initial velocity

Initial longitudinal velocity Lets you specify an initial velocity in longitudinal direction.

Initial vehicle height

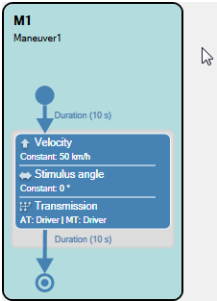
Initial height mode Lets you select the reference point for specifying the initial height.

Option	Description
Relative to road	Lets you specify the initial vehicle height relative to the road.
Offset to tire radius	Lets you specify the initial vehicle height as an offset to the tire radius.

Initial height value Lets you specify the initial height of the ASM vehicle in meters. This value is relative to the road or tire depending on the setting of the Initial height mode property.

Consistency	Valid Displays the consistency state. If the specified properties are not consistent, the reasons for the inconsistency are given.
Related topics	<div>Basics<div>Basics of Maneuvers..... 27</div></div> <div>HowTos<div>How to Start Scenario Creation..... 41</div></div> <div>References<div>Traffic Scenario Properties..... 166</div></div>

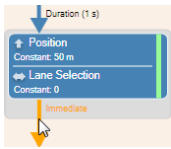
Traffic Scenario Properties

Purpose	To specify the properties of a scenario.
Access	<div>To access the properties, click the workspace on the Scenario pane outside any element.</div> <div></div>

Traffic scenario properties	<div>Author Lets you specify the name of the author.</div> <div>Creation date Displays the date on which the scenario has been created.</div> <div>Description Lets you specify a description for the scenario.</div> <div>Name Displays the name of the scenario. The name is specified when you save the scenario by using the Save As command.</div>
-----------------------------	---

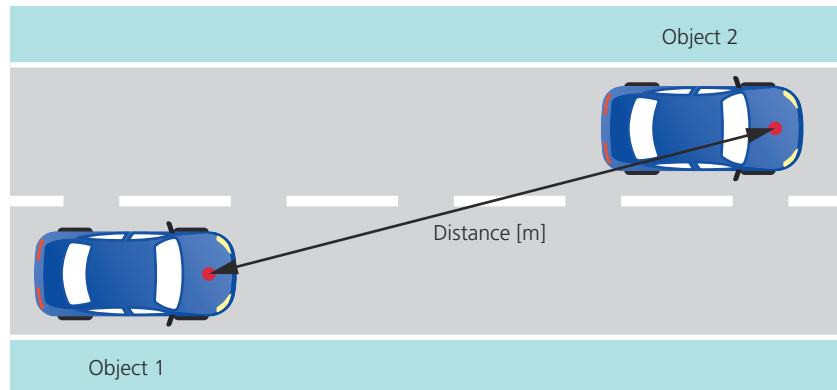
References properties	<p>Change road Lets you select another road for the scenario.</p> <p>Reload road Lets you reload the referenced road. Click this button to update the routes and position markers specified for the selected road.</p> <p>Remove road Lets you remove the reference to the road.</p> <p>Road Displays the name of the road that is selected for the scenario.</p>
Consistency	<p>Consistency Displays the consistency state.</p>
Related topics	<p>HowTos</p> <p>How to Start Scenario Creation..... 41</p> <p>References</p> <p>Save As..... 89</p>

Transition Properties

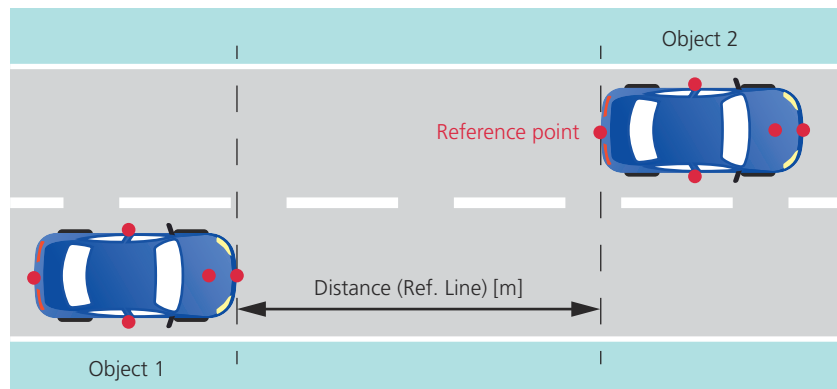
Purpose	To specify the properties of a transition.
Access	<p>To access the properties, click the transition (name or arrow) on the Scenario pane.</p> 

Transition properties	<p>Add a New Condition Lets you add a new end condition to the transition. A condition type can only be used once in a transition. The following end condition types are possible:</p> <p>Duration: The next segment is started after a specified duration.</p> <p>Endless: The active segment is executed endlessly.</p> <p>Immediate: The next segment is started immediately.</p> <p>Position: The next segment is started when a specific specified vehicle has a specified position.</p> <p>Driven Distance: The next segment is started when a specified vehicle has traveled a specified distance.</p>
------------------------------	---

Distance (between main points): The next segment is started when the distance between two specified objects (vehicles or position markers) is greater or smaller than a specified value. The distance is measured between the main points of the objects.



Distance (along road reference line): The next segment is started when the distance between two specified objects (vehicles or position markers) is greater or smaller than a specified value. The distance is measured between selectable reference points of the objects along the reference line.



Lateral acceleration: (Only for a maneuver) The next segment is started when the lateral acceleration of the ASM vehicle is greater or smaller than a specified value.


Time Gap: (Only for fellows and global user signals) The next segment is started when the time gap between two specified objects (vehicles or position markers) is greater or smaller than a specified value.

Velocity: The next segment is started when a specified vehicle has reached or exceeded a specified velocity.


Trigger: The next segment is started when an event is triggered externally.

Conditions Lets you specify the properties of the end conditions.

Condition Type	Property	Description
Duration	Duration	Lets you specify the duration of the active segment in seconds.
Endless	—	The Endless type has no properties.

Condition Type	Property	Description
Immediate	–	The Immediate type has no properties.
Position	Position	Lets you enter the position value that must be reached or exceeded to end the active segment and start the next one.
	Reference vehicle	Lets you select the ASM vehicle or a fellow as the vehicle that has to meet the specified position condition.
Driven Distance	Distance	Lets you enter the driven distance value that must be reached or exceeded to end the active segment and start the next one.
	Reference vehicle	Lets you select the ASM vehicle or a fellow as the vehicle that has to meet the specified driven distance condition.
Distance (between main points)	Approximation Type	Lets you select whether the distance between two objects must fall below or exceed the specified value to end the active segment and start the next one.
	Value	Lets you enter the value that the distance between two objects must exceed or fall below to end the active segment and start the next one.
	Object 1	Lets you select the ASM vehicle or a fellow that has to meet the specified distance condition.
	Object 2	Lets you select the vehicle (ASM vehicle, fellow) or a position marker that has to meet the specified distance condition relative to object 1. A position marker is a trigger point specified on a road or junction element. Refer to Position Markers (ModelDesk Road Creation ).
	Object 2 (Type)	Lets you select the type of object 2.
Distance (along road reference line)	Value	Lets you enter the value that the distance between two objects

Condition Type	Property	Description
	Object 1	must exceed or fall below to end the active segment and start the next one. Lets you select the ASM vehicle or a fellow that has to meet the specified distance condition.
	Object 1 (reference point)	Lets you select the reference point of object 1 that is used for the condition.
	Object 2	Lets you select the vehicle (ASM vehicle, fellow) that has to meet the specified distance condition relative to object 1.
	Object 2 (reference point)	Lets you select the reference point of object 2 that is used for the condition.
	Approximation type	Lets you select whether the distance between two objects must fall below or exceed the specified value to end the active segment and start the next one.
Lateral acceleration	Value	Lets you enter the value that the velocity must exceed or fall below to end the active segment and start the next one.
	Compare	Lets you select whether the velocity must exceed or fall below the specified value to end the active segment and start the next one.
Time Gap	Approximation Type	Lets you select whether the time gap between two objects must fall below or exceed the specified value to end the active segment and start the next one.
	Object 1	Lets you select the ASM vehicle or a fellow that has to meet the specified time gap condition.
	Object 2	Lets you select the vehicle (ASM vehicle or fellow) or a position marker that has to meet the specified time gap condition relative to object 1. A position marker is a trigger point specified on a road or junction element. Refer to Position

Condition Type	Property	Description
	Object 2 (Type)	Markers (ModelDesk Road Creation ).
	Value	Lets you select the type of object 2. Lets you enter the value that the distance between two objects must exceed or fall below to end the active segment and start the next one.
Velocity	Velocity	Lets you enter the value that the velocity must reach or exceed or fall below to end the active segment and start the next one.
	Compare	Lets you select whether the velocity must exceed or fall below the specified value to end the active segment and start the next one.
	Reference vehicle	Lets you select the ASM vehicle or a fellow as the vehicle that has to meet the specified velocity condition.
Trigger	Trigger	Lets you select an external trigger to end the active segment and start the next one.

Condition delay Lets you specify a value for time delay. You must first select the Enable delay property.

Consistency Displays the consistency state. If the specified properties are not consistent, the reasons for the inconsistency are given.

Description Lets you specify a description for the transition.

Enable delay Lets you enable a delay for the transition. When the property is enabled, the next segment is started with the delay specified in the Condition delay property.

Evaluated Displays whether the appropriate condition will be evaluated in the simulation. If this property is selected, the appropriate condition is evaluated and can end the activity.

Related topics**Basics**

[Basics of Transitions.....](#) 36

HowTos

[How to Specify a Transition for a Maneuver.....](#) 53

Source Properties

Introduction

The following properties are used in several other properties to specify the kind of source.

Circle

To specify a circle source.

Radius Lets you specify the end radius of the circle source.

Direction Lets you specify the direction.

Control Mode Lets you specify the control mode.

Constant

To specify a constant source.

Constant Lets you specify the constant value.

Hold current value Lets you specify to keep the final value from the previous segment until the end of the current segment.

Pulse

To specify a pulse.

Direction Lets you specify the direction.

Steering wheel angle Lets you specify the steering wheel angle.

Step 1 start Lets you specify the start value of step 1.

Step 1 end Lets you specify the end value of step 1.

Step 2 start Lets you specify the start value of step 2.

Step 2 end Lets you specify the end value of step 2.

Interpretation type Lets you select the interpretation type.

Ramp	<p>To specify a ramp source.</p> <p>Duration Lets you specify the total time it takes to apply the ramp signal.</p> <p>End value Lets you specify the end value.</p> <p>Start value Lets you specify the start value.</p> <p>Interpretation type Lets you select the interpretation type.</p> <p>Start with current value Lets you specify to start the segment with the current value.</p>
Sine	<p>To specify a sine source.</p> <p>Start amplitude Lets you specify the amplitude at the beginning of the segment.</p> <p>Duration Lets you specify the total time it takes to apply the sine signal.</p> <p>End frequency Lets you specify the frequency at the end of the segment.</p> <p>Start frequency Lets you specify the frequency at the beginning of the segment.</p> <p>End amplitude Lets you specify the amplitude at the end of the segment.</p> <p>Offset Lets you specify an offset.</p> <p>Interpretation type Lets you select the interpretation type.</p>
Sine with dwell	<p>To specify a sin with dwell source.</p> <p>Direction Lets you specify the direction.</p> <p>Steering wheel angle Lets you specify the steering wheel angle.</p> <p>Frequency Lets you specify the frequency.</p> <p>Dwell Lets you specify the dwell.</p> <p>Interpretation type Lets you specify the interpretation type.</p>
Step	<p>To specify a step source.</p> <p>Direction Lets you specify the direction.</p> <p>Interpretation type Lets you specify the interpretation type.</p> <p>Steering wheel angle Lets you specify the steering wheel angle.</p> <p>Value 1 Lets you specify the first value.</p> <p>Value 2 Lets you specify the second value.</p>

Table

To specify a table source.

Edit table Lets you edit the table.

Interpretation type Lets you specify the interpretation type.

User signal

To specify a user signal.

Name Displays the name of the user signal. You can specify the name in the Scenario Settings dialog.

Index Displays the index of the user signal.

Valid Displays the information whether the user signal is valid in the scenario.

Delay Lets you specify the duration after which the value is applied.

Type Lets you select the type of the user signal.

Parameters Lets you specify the parameters of the selected source type.

Related topics**References**


Clutch and Gear (MT) Properties.....	122
Lateral Type Properties (Maneuver).....	133
Longitudinal Type Properties (Maneuver).....	151
Scenario Settings Dialog.....	100

Maneuver Control Commands

Introduction ModelDesk provides commands for controlling the maneuver.

Where to go from here	Information in this section
	Reset..... 175 To reset the active maneuver.
	Start..... 176 To start the active maneuver.
	Stop..... 177 To stop the running maneuver.

Reset

Access	You can access this command via:	
	Ribbon	Home – Maneuver Control
	Context menu of	None
	Shortcut key	None
	Icon	

Purpose To reset the active maneuver.




Result The active maneuver is reset.

Description The command is available only if the simulation model contains the RESET block of the ASM_Utils_lib library.

Related topics

Basics	
Basics of Controlling Maneuvers and Driving Cycles.....	74
HowTos	
How to Control a Maneuver or a Driving Cycle.....	76
References	
Start.....	176
Stop.....	177

Start

Access	<p>You can access this command via:</p> <table><tr><td>Ribbon</td><td>Home – Maneuver Control</td></tr><tr><td>Context menu of</td><td>None</td></tr><tr><td>Shortcut key</td><td>None</td></tr><tr><td>Icon</td><td></td></tr></table>	Ribbon	Home – Maneuver Control	Context menu of	None	Shortcut key	None	Icon	
Ribbon	Home – Maneuver Control								
Context menu of	None								
Shortcut key	None								
Icon									
Purpose	To start the active maneuver.								
Result	The active maneuver is started.								
Description	The command is available only if the simulation model contains the MANEUVER_START block of the ASM_Utils_lib library.								

Related topics**Basics**

[Basics of Controlling Maneuvers and Driving Cycles.....](#) 74

HowTos

[How to Control a Maneuver or a Driving Cycle.....](#) 76

References

[Reset.....](#) 175

[Stop.....](#) 177

Stop

Access

You can access this command via:

Ribbon	Home – Maneuver Control
Context menu of	None
Shortcut key	None
Icon	

Purpose

To stop the running maneuver.

Result

The maneuver is stopped.

Description

The command is available only if the simulation model contains the MANEUVER_STOP block of the ASM_Utils_lib library.

Related topics

Basics	
Basics of Controlling Maneuvers and Driving Cycles.....	74
HowTos	
How to Control a Maneuver or a Driving Cycle.....	76
References	
Reset.....	175
Start.....	176

Automation

Where to go from here

Information in this section

Programming ModelDesk Automation.....	180
Classes for Maneuver Control.....	193
Classes for Creating Scenarios.....	198
Classes for Configuring the Maneuver.....	211
Classes for Configuring Fellows.....	240
Classes for Configuring Global User Signals.....	265
Classes for Configuring Conditions.....	282
Classes for Model Configuration.....	298
Classes for Configuring Traffic Drivers.....	310
Miscellaneous Classes for Scenarios.....	321
Constants for Scenarios.....	385

Programming ModelDesk Automation

Where to go from here

Information in this section

Automating Scenarios in Python.....	180
You can change the parameters of the scenario.	
Automated Maneuver Control in Python.....	184
You can start, stop, and reset maneuver.	
Overview of the Classes for Creating Scenarios.....	187
The overview gives you a quick overview of the classes for the Scenario Editor.	
Overview of the Object Model for Model Configuration.....	191
The overview gives you a quick overview of the classes for the setting the model configuration.	
Overview of the Object Model for Configuring Traffic Drivers.....	192
This gives you a quick overview of the classes for for configuring traffic drivers.	

Automating Scenarios in Python

Introduction

You can change the parameters of the fellow vehicles in a scenario.

Compatibility of the automation interface

You can use scripts that use the tool automation of previous ModelDesk versions. In ModelDesk 4.7, the Traffic Editor has been renamed to Scenario Editor because it also supports maneuvers. However, scripts that use the tool automation of ModelDesk 4.1 up to ModelDesk 4.6 can still be used.

Accessing the experiment

The following example shows how you can open a project and activate an experiment. Replace the project name and path by your own project in the script below.

```
from win32com.client import Dispatch
# Start ModelDesk, Load project and activate experiment
Application = Dispatch("ModelDesk.Application")
Application.Visible = True
MyProject = Application.OpenProject(r"C:\ExamplePath\Example_001\Example_001.CDP")
MyExperiments = MyProject.Experiments
MyExperiment = MyExperiments.Item(0)
MyActiveExperiment = MyExperiment.Activate(False)
```

A project is opened and an experiment is accessed in this part. For a description, refer to [Handling Projects and Experiments in Python \(ModelDesk Project and Experiment Management !\[\]\(8af806fb1314382d09bc5ec5b767526c_img.jpg\)](#)).

Creating a new scenario

The following listing shows how to create a scenario.

```
import dspace.com
Enums = dspace.com.Enums(Application)
# To create a new scenario file
MyProject.Pool.CreateFile(Enums.ContentTypes.Scenario, 'MyNewScenario', True)
```

The scenario is created in the pool of the project. To work with the scenario, you must activate it.

Activating a scenario

The following listing shows how to activate a scenario.

```
# Activate the scenario
MyActiveExperiment.ActivateTrafficScenario("Junction_TurnLeft_1F_Crossing")
# Get an object for handling the scenario
MyScenario = MyActiveExperiment.TrafficScenario
# Print information on the scenario
print "Name: ", MyScenario.Name
print "Comment: ", MyScenario.Comment
print "Author: ", MyScenario.Author
print "Corresponding road: ", MyScenario.Road
```

This part activates a scenario. The `MyScenario` object is created for accessing the scenario. Information about the scenario is printed.

Accessing and specifying the maneuver

The following listing shows how to access the maneuver of a scenario.

```
# Get object for the maneuver
MyManeuver = MyScenario.Maneuver.Item(0)
print "Name of the maneuver: ", MyManeuver.CompleteName
```

The following part shows how you can add a new sequence to the maneuver. It modifies the first segment that is added by default.

```
# Add a new sequence
MySequence = MyManeuver.Sequences.Add()
# Add an Immediate condition
MySequence.StartCondition.Conditions.Add("Immediate")
# Remove the first condition, it is not evaluated
MySequence.StartCondition.Conditions.Remove(0)
# Set first activity to the Standstill Longitudinal profile
MySegment = MySequence.Segments.Item(0)
MySegment.Activity.LongitudinalType.Activate("Standstill")
MySegment.Activity.LateralType.Activate("NotUsed")
MySegment.Transition.Conditions.Item(0).Duration = 3
```

The following part shows how you can stimulate a pedal. The acceleration pedal is stimulated by a ramp first. Afterwards, the pedal position is held for 10 seconds.

```
# Activity to stimulate the accelerator pedal
# Pedal is set from 0 to 70% in 1 second
MySegment = MySequence.Segments.Append()
MyLongType = MySegment.Activity.LongitudinalType
MyLongType.Activate("PedalStimulus")
# Activate the Ramp source type
MyPedalSet = MyLongType.ActiveElement.AccelerationPedal.Activate("Ramp")
MyPedalSet.Start = 0
MyPedalSet.End = 70
MyPedalSet.Extent = 1
MySegment.Transition.Conditions.Item(0).Duration = 1
# Activity to hold the acceleration pedal
MySegment = MySequence.Segments.Append()
MyLongType = MySegment.Activity.LongitudinalType
MyLongType.Activate("PedalStimulus")
# Activate the Constant source type
MyPedalSet = MyLongType.ActiveElement.AccelerationPedal.Activate("Constant")
MyPedalSet.HoldCurrentValue = True
MySegment.Transition.Conditions.Item(0).Duration = 10
```

Adding and specifying fellows

The following listing shows how to access the Fellows collection.

```
# Get object for scenario fellows
MyFellows = MyScenario.Fellows
print "Number of scenario fellows: ", MyFellows.Count
```

The **MyFellows** object is created for accessing the fellows. The number of fellows is printed.

```
# Add a new fellow to the scenario
MyFellow = ActiveTraffic.Fellows.Add()
# Specify a name for the fellow
MyFellow.Name = 'MyNewFellow'
# Select a traffic object for the fellow
MyFellow.TrafficObjectType.Activate('Vehicles\\SimpleCarBrown.tro')
```

A fellow is created with one sequence containing two segments. The first segment is used to initiate the fellow's position, i.e. the activity has the **Position** type and the transition has the **Immediate** type. The following listing shows how to access the first segment and specify a new position value.

```
# Access the fellow's 1st sequence and segment
MySequence = MyFellow.Sequences.Item(0)
MySegment = MySequence.Segments.Item(0)
# Access the activity of the 1st segment
ActiveLongitudinal = MySegment.Activity.LongitudinalType.ActiveElement
# Specify the position value
ActiveLongitudinal.SourceType.ActiveElement.Constant = 42
```

The following listing demonstrates how to append a segment to the fellow's sequence and specify the segment's activity and transition.

```
# Add a segment
MySegment = MySequence.Segments.Append()
# Specify the activity of the segment
ActiveLongitudinal = MySegment.Activity.LongitudinalType.Activate("StopWithin")
ActiveLongitudinal.SourceType.ActiveElement.HoldCurrentValue = True
ActiveLongitudinal.SourceType.ActiveElement.StopDistance = 42
ActiveLateral = MySegment.Activity.LateralType.Activate("Continue")
```

```
# Access the 1st condition. It is created by default.
Transition = MySegment.Transition.Conditions.Item(0)
Transition.Duration = 8
# Add a Position condition and specify its properties
Transition = MySegment.Transition.Conditions.Add(4)
Transition.Position = 54
```

Adding and specifying global user signals

A user signal is created with one sequence and segment. In the activity of the segment a constant value is set, the transition has the `Duration` type by default. The following listing shows how to add a global user signal to a scenario and specify the value and duration.

```
# Add a user signal to the collection
MyUserSignal = ActiveTraffic.UserSignals.Add()
# Specify a name for the global user signal
MyUserSignal.Name = 'MyUserSignal'
# Access the 1st sequence and segment of the user signal
MySequence = MyUserSignal.Sequences.Item(0)
MySegment = MySequence.Segments.Item(0)
# Specify a value and duration time
MySegment.Activity.ActiveElement.Constant = 42
MySegment.Transition.Conditions.Item(0).Duration = 8
```

Configuring traffic drivers

The following listing shows how you can create a new traffic driver in the pool.

```
import dspace.com
Enums = dspace.com.Enums(Application)
# To create a new traffic driver
MyProject.Pool.CreateFile(Enums.ContentTypes.TrafficDriver, 'MyNewTrafficDriver', True)
```

The following listing shows how you can configure a traffic driver.

```
# Access the traffic drivers of the current project
MyTrafficDrivers = MyProject.TrafficDrivers
# Access a specific traffic driver
MyTrafficDriver = MyTrafficDrivers.Item(0)
# Set properties
MyTrafficDriver.DrivingStyle.DetectionRange = 110.0
MyTrafficDriver.RuleCompliantDriving = DistanceAtStandstill = 0.5
MyTrafficDriver.TrafficSignRecognition.DistanceFront = 42.0
# Save the traffic driver
MyTrafficDriver.SaveAs('MyTrafficDriver.trd', 0)
```

The following listing shows how you can use the traffic driver for the maneuver.

```
# Activate the traffic driver for the maneuver
MyManeuver.TrafficDriverType.Activate('MyTrafficDriver.trd')
```

Saving and downloading the scenario

The following listing shows how to save and download the scenario.

```
# Save the scenario
MyScenario.Save()
# Check the scenarios consistency
MyScenario.CheckConsistency()
# Download the scenario
MyScenario.Download()
```

After saving the scenario, the scenario is check for consistency.

Related topics

Basics

[Handling Projects and Experiments in Python \(ModelDesk Project and Experiment Management !\[\]\(17acf1afa8cdf0b67c53d4865a5ed469_img.jpg\)](#))

References

Classes for Configuring Fellows.....	240
Classes for Configuring Global User Signals.....	265
Classes for Configuring the Maneuver.....	211
Classes for Configuring Traffic Drivers.....	310
Classes for Creating Scenarios.....	198
Constants for Scenarios.....	385
Miscellaneous Classes for Scenarios.....	321
Overview of the Classes for Creating Scenarios.....	187

Automated Maneuver Control in Python

Introduction

You can start, stop, and reset maneuvers.

Accessing the experiment

The following example shows how you can open a project and activate an experiment. Replace the project name and path by your own project in the script below.

```
from win32com.client import Dispatch
# Start ModelDesk, load project and activate experiment
Application = Dispatch("ModelDesk.Application")
Application.Visible = True
MyProject = Application.OpenProject(r"C:\ExamplePath\Example_001\Example_001.CDP")
MyExperiments = MyProject.Experiments
MyExperiment = MyExperiments.Item(0)
MyActiveExperiment = MyExperiment.Activate(False)
```

A project is opened and an experiment is accessed in this part. For a description, refer to [Handling Projects and Experiments in Python \(ModelDesk Project and Experiment Management !\[\]\(4146d17f71dced09c6ad789cacceaa6d_img.jpg\)](#)).

Activating a maneuver

The following listing shows how to activate a maneuver.

```
# Activate the maneuver
MyActiveExperiment.ActivateManeuver("FollowRoad")
# Get an object for handling the road
MyManeuver = MyActiveExperiment.Maneuver
# Print information on the maneuver
print "Name of the maneuver:", MyManeuver.Name
print "Comment of the maneuver:", MyManeuver.Comment
print "Author of the maneuver:", MyManeuver.Author
```

This part activates a maneuver. The `MyManeuver` object is created for accessing the maneuver. The name, comment and author of the maneuver are printed.

Starting the maneuver

The following listing shows how to start the maneuver.

```
# Start the maneuver
MyActiveExperiment.ManeuverControl.Start()
```

Example

The following example shows how you can start two maneuvers one after the other. Both maneuvers are available in the standard installation of ModelDesk. When you use the Python script, you must only replace the `ModelDeskProjectFile` and `ExperimentName` variables by your project and experiment name. Before the Python script is started, you must have access to the platform where the simulation of the appropriate simulation model should be started. When you want to monitor the maneuver, you can use MotionDesk.

```
import sys, os
import time
import win32com.client
ModelDeskProjectFile = r"C:\ExamplePath\ExampleProjectName.CDP"
ExperimentName       = r"ExampleExperimentName"
ManeuverNames        = ["MueSplit", "LaneChange"]
gManeuverStateMapping = {2: 'Init',
                        3: 'Run',
                        4: 'Stopping',
                        5: 'Wait',
                        6: 'Init Manual',
                        7: 'Manual'}

# Timeout for wait time maneuver running, stopping in seconds
Timeout = 120
if __name__ == "__main__":
    ModelDeskApplication = None
    print "Start demo: %s.\n" % time.asctime()
    try:
        #-----
        # Access ModelDesk Application
        #-----
        ModelDeskApplication = win32com.client.Dispatch("ModelDesk.Application")
        ModelDeskApplication.Visible = True
```

```

#-----
# Load and activate project
#-----
print "Initializing ModelDesk...\n"
SaveActiveProjectFlag = True
ModelDeskApplication.OpenProject(ModelDeskProjectFile, "", SaveActiveProjectFlag)
# Activate first experiment
MyActiveManeuver = ModelDeskApplication.ActiveProject.Experiments[0].Activate(False)
for ManeuverName in ManeuverNames:
    print "About to activate maneuver: '%s'" % ManeuverName
    MyActiveManeuver.ActivateManeuver(ManeuverName)
    print "Maneuver '%s' is active. Download maneuver." % MyActiveManeuver.Maneuver.Name
    MyActiveManeuver.Maneuver.Download()
    # Reset vehicle
    print "Reset vehicle..."
    MyActiveManeuver.ManeuverControl.Reset()
    # Start maneuver using ModelDesk tool automation
    print "Start maneuver..."
    MyActiveManeuver.ManeuverControl.Start(True)
    # Wait for maneuver starts running.
    ManeuverRunning = False
    StartTime = time.time()
    while(True):
        if time.time() - StartTime >= Timeout:
            print "TIMEOUT REACHED! Break polling loop to wait for maneuver start .."
            # Stop maneuver
            MyActiveManeuver.ManeuverControl.Stop()
            break
        if gManeuverStateMapping[MyActiveManeuver.ManeuverControl.ManeuverState] == "Run":
            print "Maneuver is running..."
            ManeuverRunning = True
            break
        # Wait 1 second
        time.sleep(5)
    # Wait for end of maneuver run.
    StartTime = time.time()
    if ManeuverRunning:
        while(True):
            if time.time() - StartTime >= Timeout:
                print "TIMEOUT REACHED. Break polling loop and stop maneuver.."
                # Stop maneuver
                MyActiveManeuver.ManeuverControl.Stop()
                break
            if gManeuverStateMapping[MyActiveManeuver.ManeuverControl.ManeuverState] != "Run":
                print "Maneuver is no longer running.."
                break
            # Wait 1 second
            time.sleep(1)
    print
    print "Demo successfully finished: %s.\n" % time.asctime()
finally:
    #-----
    # make sure to shut down of ModelDesk
    #-----
    if ModelDeskApplication != None:
        ModelDeskApplication.Quit(False)
        ModelDeskApplication = None

```

Related topics**Basics**

[Handling Projects and Experiments in Python \(ModelDesk Project and Experiment Management !\[\]\(4729e517bc6a7cd81c8025b9646574fb_img.jpg\)\)](#)















References





















[Overview of the Object Model for Accessing ModelDesk Experiments \(ModelDesk Project and Experiment Management !\[\]\(a03a7eb2f4046e1d3c76772003e549ea_img.jpg\)\)](#)

Overview of the Classes for Creating Scenarios

Classes

The following tables give you a quick overview of the classes for creating scenarios.










Class	Level
ActiveTrafficScenario on page 198	
Maneuvers on page 220	
Maneuver on page 219	
ManeuverSequences on page 231	
ManeuverSequence on page 229	
ManeuverSegments on page 225	
ManeuverSegment on page 224	
UserOutputs on page 262	
UserOutput on page 264	
UserSignalValue on page 280	
ManeuverActivity on page 223	
LongitudinalProfile on page 218	
Velocity on page 382	
Stop on page 373	
PedalStimulus on page 349	
Standstill on page 371	
LateralAcceleration on page 338	
LateralProfile on page 216	

Class	Level
NotUsed on page 348	
StimulusAngle on page 371	
StimulusTorque on page 372	
BasicRoad on page 328	
FollowRoad on page 337	
GearProfile on page 214	
SelectorProfile on page 236	
ManeuverTransition on page 234	
TransitionsConditions on page 293	
ConditionDuration on page 285	
ConditionEndless on page 286	
ConditionImmediate on page 287	
ConditionPosition on page 288	
ConditionDrivenDistance on page 284	
ConditionDistance on page 283	
ConditionVelocity on page 290	
ConditionTrigger on page 290	
ConditionLateralAcceleration on page 287	
Fellows on page 242	
Fellow on page 241	
FellowSequences on page 251	
FellowSequence on page 251	
FellowSegments on page 246	
FellowSegment on page 246	
UserOutputs on page 262	
UserOutput on page 264	
UserSignalValue on page 280	
FellowActivity on page 242	
LongitudinalProfile on page 258	
NotUsed on page 348	
Continue on page 292	
Acceleration on page 323	
AccelerationSource on page 324	
DistanceMeter on page 332	
DistanceMeterSource on page 333	
DistanceTime on page 334	
DistanceTimeSource on page 335	
Position on page 350	

Class	Level
PositionSource on page 351	10
Velocity on page 382	9
VelocitySource on page 383	10
StopWithin on page 374	9
StopWithinSource on page 374	10
LateralProfile on page 256	8
Continue on page 291	9
NotUsed on page 348	
LateralDeviation on page 339	
LateralDeviationSource on page 339	10
LaneSelection on page 341	9
LaneSelectionSource on page 342	10
FellowTransition on page 255	7
TransitionsConditions on page 293	8
ConditionDistance on page 283	9
ConditionDrivenDistance on page 284	
ConditionDuration on page 285	
ConditionEndless on page 286	
ConditionImmediate on page 287	
ConditionPosition on page 288	
ConditionTimegap on page 289	
ConditionTrigger on page 290	
ConditionVelocity on page 290	
FellowTransition on page 255 (StartCondition)	4
RouteSelection on page 358	
Route on page 357	5
TrafficObjectTypes on page 260	3
TrafficObjectType on page 261	4
GlobalUserSignals on page 266	1
GlobalUserSignal on page 269	2
UserSignalSequences on page 275	3
UserSignalSequence on page 278	4
UserSignalSegments on page 270	5
UserSignalSegment on page 274	6
UserSignalValue on page 280	7
UserSignalTransition on page 279	
UserSignalSegment on page 274 (Successor)	

Class	Level
UserSignalTransition on page 279 (StartCondition)	

The following tables give you a quick overview of the miscellaneous classes for creating scenarios.

Class	Level
ApproximationOperators on page 326	
ApproximationOperator on page 327	
DependencyTypes on page 329	
DependencyTypeAbsolute on page 330	
DependencyTypeRelative on page 331	
RelativeOperators on page 353	
RelativeOperators on page 352	
RelativeVehicleSelection on page 355	
RelativeObject on page 355	
SourceTypes on page 360	
Constant on page 362	
Extern on page 362	
FinalDeviation on page 362	
FinalDistance on page 363	
FinalLane on page 363	
FinalVelocity on page 364	
LaneChange on page 365	
Ramp on page 366	
Sine on page 367	
SmoothLaneChange on page 368	
StopWithin on page 369	
SimpleConstant on page 366	
YawAngleConstant on page 370	
Table on page 376	
Step on page 368	
Pulse on page 365	
SineWithDwell on page 367	
Circle on page 361	
UntilStop on page 370	
Fixed on page 364	
Straight on page 369	

Related topics

Basics

[Automating Scenarios in Python.....](#) 180









References

[Classes for Configuring Fellows.....](#) 240
[Classes for Configuring Global User Signals.....](#) 265
[Classes for Configuring the Maneuver.....](#) 211
[Classes for Creating Scenarios.....](#) 198
[Constants for Scenarios.....](#) 385
[Miscellaneous Classes for Scenarios.....](#) 321

Overview of the Object Model for Model Configuration

Classes

The following tables give you a quick overview of the classes for the setting the model configuration.

Classes	Level
ActiveProject	
ModelConfiguration on page 304	
ManeuverConfiguration on page 303	
ReadOnlyListElements on page 306	
Triggers	
UserSignals	
ListElement on page 302	
ScenarioConfiguration on page 308	
ListElements on page 298	
FellowUserSignals	
GlobalUserSignals	
Triggers	
ListElement on page 302	

Related topics





References

[Environment Configuration Dialog.....](#) 92
[Project \(ModelDesk Project and Experiment Management !\[\]\(c444627dab9fee9a1550c053ffaaaae2_img.jpg\)\)](#)

Overview of the Object Model for Configuring Traffic Drivers

Classes

The following tables give you a quick overview of the classes for configuring traffic drivers.

Classes	Level
ActiveProject	
TrafficDrivers on page 313	
TrafficDriver on page 316	
DrivingStyle on page 310	
RuleCompliantDriving on page 311	
TrafficSignRecognition on page 319	

Related topics

References

Project (ModelDesk Project and Experiment Management )	
Traffic Driver Dialog.....	101

Classes for Maneuver Control

Where to go from here

Information in this section

[ManeuverControl..... 193](#)
To control the active maneuver.

Information in other sections

[Overview of the Classes for Creating Scenarios..... 187](#)
The overview gives you a quick overview of the classes for the Scenario Editor.

[Automating Scenarios in Python..... 180](#)
You can change the parameters of the scenario.

ManeuverControl

Purpose

To control the active maneuver.

Where to go from here

Information in this section

[Class Description \(ManeuverControl\)..... 194](#)
To describe the class and its attributes.

[Start..... 195](#)
To start the active maneuver.

[Reset..... 196](#)
To reset the active maneuver.

[Stop..... 196](#)
To stop the running maneuver.

Information in other sections

[Automated Maneuver Control in Python..... 184](#)
You can start, stop, and reset maneuver.

Class Description (ManeuverControl)

Syntax

```
ManeuverControl = Experiment.ManeuverControl
```

Purpose

To control the active maneuver.

Description

So that you can change the state of the maneuver, the maneuver must be in specific states. These states depend on the kind of model. Refer to the state diagram of the maneuver.



Vehicle dynamics: [Maneuver Scheduler \(ASM Environment Reference !\[\]\(3211b5d1d968fc1665909b34f9f16010_img.jpg\)](#))

Engine: [Maneuver Control \(ASM Drivetrain Basic Reference !\[\]\(6059a5aa8b4ca7bb793408023d6c6e42_img.jpg\)](#))

You can read the current state with the **ManeuverState** attribute.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ManeuverState	Long	<p>To get the maneuver state. The meanings of the values depend on the kind of model.</p> <p>Maneuver state for vehicle dynamics (refer to Maneuver Scheduler (ASM Environment Reference )):</p> <ul style="list-style-type: none"> ▪ 1: Maneuver starting ▪ 2: Maneuver initializing ▪ 3: Maneuver running ▪ 4: Maneuver braking ▪ 5: Maneuver waiting ▪ 6: Manual initializing ▪ 7: Manual driving <p>Maneuver state for engine (refer to Maneuver Control (ASM Drivetrain Basic Reference )):</p> <ul style="list-style-type: none"> ▪ 1: Maneuver stopped ▪ 2: Maneuver paused ▪ 3: Maneuver running ▪ 4: Manual state

Methods

The class contains the following methods:

Method	Purpose
Reset	To reset the active maneuver. Refer to Reset on page 196.
Start	To start the active maneuver. Refer to Start on page 195.
Stop	To stop the running maneuver. Refer to Stop on page 196.

Related topics**Basics**

Automated Maneuver Control in Python..... 184
 Basics of Controlling Maneuvers and Driving Cycles..... 74

References

ActiveExperiment (ModelDesk Project and Experiment Management )

Start

Class

ManeuverControl

Syntax

```
RetVal = ManeuverControl.Start(DownloadExperimentIfApplicationNotRunning)
```

Purpose

To start the active maneuver.

Parameters

The method uses the following parameters:

Parameter	Type	Description
DownloadExperimentIfApplicationNotRunning	Boolean	If True: If no or another simulation application is loaded on the simulation platform, the simulation application of the ModelDesk experiment and the active experiment is loaded. The active experiment contains the scenario, road, and parameter set.

Return value

The method returns an object of the following type:

Type	Description
Boolean	True if the maneuver is successfully started

Related topics**References**

Class Description (ManeuverControl)..... 194

Reset

Class ManeuverControl

Syntax `RetVal = ManeuverControl.Reset()`

Purpose To reset the active maneuver.

Parameters –

Return value The method returns an object of the following type:

Type	Description
Boolean	True if the maneuver is successfully reset

Related topics

References

[Class Description \(ManeuverControl\)..... 194](#)

Stop

Class ManeuverControl

Syntax `RetVal = ManeuverControl.Stop()`

Purpose To stop the running maneuver.

Parameters –

Return value The method returns an object of the following type:

Type	Description
Boolean	True if the maneuver is successfully stopped

Related topics

References

Class Description (ManeuverControl).....	194
--	-----

Classes for Creating Scenarios

Where to go from here

Information in this section

ActiveTrafficScenario.....	198
The scenario currently linked to the experiment.	
DriverProperties.....	206
To get the properties of the driver.	
ManeuverProperties.....	207
To get the maneuver properties.	
ScenarioProperties.....	208
To get information of the properties of a scenario.	
TrafficProperties.....	210
To get the traffic properties.	

Information in other sections

Overview of the Classes for Creating Scenarios.....	187
The overview gives you a quick overview of the classes for the Scenario Editor.	
Automating Scenarios in Python.....	180
You can change the parameters of the scenario.	

ActiveTrafficScenario

Purpose

This is the scenario currently linked to the experiment.

Where to go from here

Information in this section

Class Description (ActiveTrafficScenario).....	199
To describe the class and its attributes.	
CheckConsistency.....	200
To check the complete scenario for consistency.	
Download.....	201
To download the active scenario to Simulink or the real-time hardware.	

DownloadContent.....	202
To download the maneuver and/or traffic to the real-time hardware or Simulink.	
GenerateModelIni.....	202
To generate the model initialization file.	
RemoveRoad.....	203
To reload the road assigned to the scenario.	
RemoveRoad.....	204
To remove the road connection from the scenario.	
ReplaceContent.....	204
To replace a part of the scenario, for example, the settings of the maneuver of another scenario.	
Save.....	205
To save the active scenario.	
SaveAs.....	205
To save the active scenario under a new name.	

Class Description (ActiveTrafficScenario)

Syntax

```
TrafficScenario = ActiveExperiment.TrafficScenario
```

Purpose

To modify the contents of the active scenario.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Author	String	To get/set the author of the scenario.
Comment	String	To get/set the comment of the scenario.
Fellows	Fellows ¹⁾	To get the Fellows collection object of the scenario.
Maneuver	Maneuvers ²⁾	To get the maneuver of the scenario.
Name	String	To get the name of the scenario.
Properties	ScenarioProperties ³⁾	To get the properties of the scenario.
Road	String	To get the name of the road assigned to the scenario.
UserSignals	GlobalUserSignals ⁴⁾	To get the GlobalUserSignals collection object of the scenario.

¹⁾ Refer to [Fellows](#) on page 242.

²⁾ Refer to [Maneuvers](#) on page 220.

³⁾ Refer to [ScenarioProperties](#) on page 208.

Attributes	Type	Purpose
4) Refer to GlobalUserSignals on page 266.		

Methods

The class contains the following methods:

Method	Purpose
CheckConsistency	To check the consistency of the complete scenario. Refer to CheckConsistency on page 200.
Download	To download the active scenario to the real-time hardware or Simulink. Refer to Download on page 201.
DownloadContent	To download the maneuver and/or traffic to the real-time hardware or Simulink. Refer to DownloadContent on page 202.
GenerateModelIni	To generate the model initialization file. Refer to GenerateModelIni on page 202.
RemoveRoad	To reload the road assigned to the scenario. Refer to RemoveRoad on page 203.
RemoveRoad	To remove the road connection from the scenario. Refer to RemoveRoad on page 204.
ReplaceContent	To replace a part of the scenario, for example, the settings of the maneuver of another scenario. Refer to ReplaceContent on page 204.
Save	To save the active scenario with its current name. Refer to Save on page 205.
SaveAs	To save the active scenario with a specified name. Refer to SaveAs on page 205.

Related topics

References

[ActiveExperiment \(ModelDesk Project and Experiment Management !\[\]\(17acf1afa8cdf0b67c53d4865a5ed469_img.jpg\)](#))

CheckConsistency

Class

ActiveTrafficScenario

Syntax

```
ActiveTrafficScenario.CheckConsistency()
```

Purpose

To check the complete scenario for consistency.

Parameters	—				
Return value	<p>The method returns a value of the following type:</p> <table> <tr> <th>Type</th><th>Description</th></tr> <tr> <td>Boolean</td><td>True if the scenario is consistent, otherwise false.</td></tr> </table>	Type	Description	Boolean	True if the scenario is consistent, otherwise false.
Type	Description				
Boolean	True if the scenario is consistent, otherwise false.				
Related topics	<p>References</p> <p>Class Description (ActiveTrafficScenario)..... 199</p>				

Download

Class	ActiveTrafficScenario				
Syntax	<code>ActiveTrafficScenario.Download()</code>				
Purpose	To download the active scenario to Simulink or the real-time hardware.				
Parameters	—				
Return value	<p>The method returns a value of the following type:</p> <table> <tr> <th>Type</th><th>Description</th></tr> <tr> <td>Boolean</td><td>True if successful, otherwise false.</td></tr> </table>	Type	Description	Boolean	True if successful, otherwise false.
Type	Description				
Boolean	True if successful, otherwise false.				
Related topics	<p>References</p> <p>Class Description (ActiveTrafficScenario)..... 199</p>				

DownloadContent

Class ActiveTrafficScenario

Syntax `ActiveTrafficScenario.DownloadContent(DownloadSelectionTypes Type)`

Purpose To download the maneuver and/or traffic to the real-time hardware or Simulink.

Parameters The method uses the following parameters:

Parameter	Type	Description
Type	DownloadSelectionTypes ¹⁾	Lets you specify whether the maneuver, the traffic, or the whole scenario is downloaded.

¹⁾ Refer to [DownloadSelectionTypes](#) on page 386.

Return value The method returns a value of the following type:

Type	Description
Boolean	True if successful, otherwise false.

Related topics

References

[Class Description \(ActiveTrafficScenario\)](#)..... 199

GenerateModelIni

Class ActiveTrafficScenario

Syntax `ActiveTrafficScenario.GenerateModelIni(string FilePath)`

Purpose To generate the model initialization file.

Parameters

The method uses the following parameters:

Parameter	Type	Description
FilePath	String	The target file path of the model initialization file.

Return value

The method returns a value of the following type:

Type	Description
Boolean	True if the file could be created, otherwise false.

Related topics**References**

[Class Description \(ActiveTrafficScenario\)..... 199](#)

RemoveRoad

Class

ActiveTrafficScenario

Syntax

```
RetVal = ActiveTrafficScenario.ReloadRoad()
```

Purpose

To reload the road assigned to the scenario.

Parameters

—

Return value

The method returns a value of the following type:

Type	Description
Boolean	True if successful, otherwise false.

Related topics**References**

[Class Description \(ActiveTrafficScenario\)..... 199](#)

RemoveRoad

Class ActiveTrafficScenario

Syntax `RetVal = ActiveTrafficScenario.RemoveRoad()`

Purpose To remove the road connection from the scenario.

Parameters –

Return value The method returns a value of the following type:

Type	Description
Boolean	True if successful, otherwise false.

Related topics

References

[Class Description \(ActiveTrafficScenario\).....](#) 199

ReplaceContent

Class ActiveTrafficScenario

Syntax `ActiveTrafficScenario.ReplaceContent(FilePath, Type)`

Purpose To replace a part of the scenario, for example, the settings of the maneuver of another scenario.

Parameters The method uses the following parameters:

Parameter	Type	Description
FilePath	String	The scenario file that is used for the replacement.
Type	ReplacementTypes ¹⁾	The part of the scenario that is replaced.

¹⁾ Refer to [ReplacementTypes](#) on page 389.

Return value

The method returns a value of the following type:

Type	Description
Boolean	True if successful, otherwise false.

Related topics**References**

[Class Description \(ActiveTrafficScenario\)..... 199](#)

Save

Class

ActiveTrafficScenario

Syntax

```
ActiveTrafficScenario.Save()
```

Purpose

To save the active scenario.

Parameters

—

Return value

-

Related topics**References**

[Class Description \(ActiveTrafficScenario\)..... 199](#)

SaveAs

Class

ActiveTrafficScenario

Syntax

```
ActiveTrafficScenario.SaveAs(FileName, OverwritingExisting)
```

Purpose To save the active scenario under a new name.

Parameters The method uses the following parameters:

Parameter	Type	Description
FileName	String	The name under which the scenario is saved.
OverwriteExisting	Boolean	Saving the active scenario overwrites an existing file called FileName if true.

Return value -

Related topics

References

[Class Description \(ActiveTrafficScenario\)..... 199](#)

DriverProperties

Purpose To get the properties of the driver.

Class Description (DriverProperties)

Syntax `DriverProperties = ManeuverProperties.Driver`

Purpose To get the properties of the driver.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Slow	DriverType ¹⁾	To get the properties of the slow driver.
Normal	DriverType ¹⁾	To get the properties of the normal driver.
Fast	DriverType ¹⁾	To get the properties of the fast driver.

¹⁾ Refer to [DriverType](#) on page 213.

Methods

—

Related topics**References**

ManeuverProperties.....	207
Scenario Properties Dialog.....	98

ManeuverProperties

Purpose

To get the maneuver properties.

Class Description (ManeuverProperties)

Syntax

```
ManeuverProperties = ScenarioProperties.Maneuver
```

Purpose

To get the maneuver properties.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
UserSignals	NamedItemList ¹⁾	To get the maneuver user signals.
Trigger	NamedItemList ¹⁾	To get the triggers.
Driver	DriverProperties ²⁾	To get the driver properties.
LateralController	LateralControllerTypes ³⁾	To get/set the lateral controller.

¹⁾ Refer to [NamedItemList](#) on page 345.

²⁾ Refer to [DriverProperties](#) on page 206.

³⁾ Refer to [LateralControllerTypes](#) on page 387.

Methods

—

Related topics

References

Scenario Properties Dialog.....	98
ScenarioProperties.....	208

ScenarioProperties

Purpose

To get information of the properties of a scenario.

Where to go from here

Information in this section

Class Description (ScenarioProperties).....	208
To describe the class and its attributes.	
Reset.....	209
To reset the scenario properties.	

Class Description (ScenarioProperties)

Syntax

```
ScenarioProperties = ActiveTrafficScenario.Properties
```

Purpose

To get information of the properties of a scenario.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
GlobalProperties	GlobalPropertiesLaneIndexInterpretationTypes ¹⁾	To get/set the lane index interpretation (relative to preferred lane and driving direction = 0 or relative to center lane = 1).
Maneuver	ManeuverProperties ²⁾	To get the properties of the maneuver.

Attributes	Type	Purpose
Traffic	TrafficProperties ³⁾	To get the properties of the traffic.

¹⁾ Refer to [GlobalPropertiesLaneIndexInterpretationTypes](#) on page 387.

²⁾ Refer to [ManeuverProperties](#) on page 207.

³⁾ Refer to [TrafficProperties](#) on page 210.

Methods

The class contains the following methods:

Method	Purpose
Reset	To reset the scenario properties. Refer to Reset on page 209.

Related topics

References

Class Description (ActiveTrafficScenario)	199
Scenario Properties Dialog	98

Reset

Class ScenarioProperties

Syntax `ScenarioProperties.Reset()`

Purpose To reset the scenario properties.

Parameters —

Return value —

Related topics

References

Class Description (ScenarioProperties)	208
--	-----

TrafficProperties

Purpose To get the traffic properties.

Class Description (TrafficProperties)

Syntax `TrafficProperties = ScenarioProperties.Traffic`

Purpose To get the properties of the traffic.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
UserSignals	NamedItemList ¹⁾	To get the fellow user signals.
Trigger	NamedItemList ¹⁾	To get the triggers.

¹⁾ Refer to [NamedItemList](#) on page 345.

Methods —

Related topics

References

Scenario Properties Dialog.....	98
ScenarioProperties.....	208

Classes for Configuring the Maneuver

Where to go from here

Information in this section

Driver.....	212
To specify the Driver type.	
DriverType.....	213
To specify the values of the driver type.	
GearProfile.....	214
To get the clutch and gear type.	
LateralProfile.....	216
To specify the lateral profile of a maneuver's activity.	
LongitudinalProfile.....	218
To specify the longitudinal profile of a maneuver's activity.	
Maneuver.....	219
To get a maneuver.	
Maneuvers.....	220
To manage the maneuvers of a scenario.	
ManeuverActivity.....	223
To get a maneuver activity.	
ManeuverSegment.....	224
To get a maneuver segment.	
ManeuverSegments.....	225
To get the maneuver segments.	
ManeuverSequence.....	229
To get a sequence of a maneuver.	
ManeuverSequences.....	231
To get all the sequences of a maneuver.	
ManeuverTransition.....	234
To get the transition of a maneuver.	
OpenClutch.....	235
To specify the OpenClutch type.	
ReferenceGear.....	235
To specify a ReferenceGear type.	
SelectorProfile.....	236
To get the selector lever type.	
Stimulus.....	238
To specify a stimulus type.	

Information in other sections

Overview of the Classes for Creating Scenarios.....	187
The overview gives you a quick overview of the classes for the Scenario Editor.	
Automating Scenarios in Python.....	180
You can change the parameters of the scenario.	

Driver

Purpose To specify the Driver type.

Class Description (Driver)

Syntax `Driver = GearProfile.ActiveElement`

Purpose To specify the Driver type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the element.
Type	ClutchAndGearTypes ¹⁾	To get the type of the element.
SaturateGear	Boolean	To get/set the saturation of the gear.
Gear	Integer	To get/set the gear.

¹⁾ Refer to [ClutchAndGearTypes](#) on page 385.

Methods —

Related topics **References**

GearProfile.....	214
----------------------------------	-----

DriverType

Purpose To specify the values of the driver type.

Class Description (DriverType)

Syntax `DriverType = FollowRoad.DriverTypeValues`

Purpose To specify the values of the driver type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
BrakeThreshold	Double	To get/set the brake acceleration threshold.
LateralAcceleration	Double	To get/set the maximum lateral acceleration.
LongitudinalAcceleration	Double	To get/set the maximum longitudinal acceleration.

Methods —

Related topics

References

[FollowRoad..... 337](#)
[Scenario Properties Dialog..... 98](#)

GearProfile

Purpose To get the clutch and gear type.

Where to go from here **Information in this section**

Class Description (GearProfile).....	214
To describe the class and its attributes.	
Activate.....	215
To activate an element for the clutch and gear profile.	

Class Description (GearProfile)

Syntax `GearProfile = ManeuverActivity.ClutchAndGearType`

Purpose To get the clutch and gear type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	OpenClutch, Driver, Stimulus, or ReferenceGear ¹⁾	To get the active element.
AvailableElements	String[]	To get a list of all the available elements.

¹⁾ Refer to [OpenClutch](#) on page 235, [Driver](#) on page 212, [Stimulus](#) on page 238, or [ReferenceGear](#) on page 235.

Methods The class contains the following methods:

Method	Purpose
Activate	To activate an element for the clutch and gear profile. Refer to Activate on page 215.

Related topics**References**

[ManeuverActivity](#)..... 223

Activate

Class

GearProfile

Syntax

```
ActivatedElement = GearProfile.Activate(Type)
```

Purpose

To activate an element for the clutch and gear profile.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Type	OpenClutch, Driver, Stimulus, or ReferenceGear ¹⁾	The type to be activated.

¹⁾ Refer to [OpenClutch](#) on page 235, [Driver](#) on page 212, [Stimulus](#) on page 238, or [ReferenceGear](#) on page 235.

Return value

The method returns the following parameter:

Type	Description
OpenClutch, Driver, Stimulus, or ReferenceGear ¹⁾	The activated element.

¹⁾ Refer to [OpenClutch](#) on page 235, [Driver](#) on page 212, [Stimulus](#) on page 238, or [ReferenceGear](#) on page 235.

Related topics**References**

[Class Description \(GearProfile\)](#)..... 214

LateralProfile

Purpose To specify the lateral profile of a maneuver's activity.

Where to go from here Information in this section

Class Description (LateralProfile).....	216
To describe the class and its attributes.	
Activate.....	217
To get a specific lateral type.	

Class Description (LateralProfile)

Syntax `LateralProfile = ManeuverActivity.LateralType`

Purpose To specify the lateral profile of a maneuver's activity.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	StimulusAngle, StimulusTorque, BasicRoad, FollowRoad, or NotUsed ¹⁾	To get the active lateral type object.
AvailableElements	String[]	To get the names of all available lateral types.

¹⁾ Refer to [StimulusAngle](#) on page 371, [StimulusTorque](#) on page 372, [BasicRoad](#) on page 328, [FollowRoad](#) on page 337, or [NotUsed](#) on page 348.

Methods The class contains the following methods:

Method	Purpose
Activate	To get a specific lateral type. Refer to Activate on page 217.

Related topics**References**

[ManeuverActivity..... 223](#)

Activate

Class

LateralProfile

Syntax

```
LateralType = LateralProfileType.Activate(Type)
```

Purpose

To get a specific lateral type.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific lateral type. The value can be specified as string (name) or integer (constant type).

Return value

The method returns an object of the following type:

Type	Description
StimulusAngle, StimulusTorque, BasicRoad, FollowRoad, or NotUsed ¹⁾	A lateral type object according to the given parameter.

¹⁾ Refer to [StimulusAngle](#) on page 371, [StimulusTorque](#) on page 372, [BasicRoad](#) on page 328, [FollowRoad](#) on page 337, or [NotUsed](#) on page 348.

Related topics**References**

[Class Description \(LateralProfile\)..... 216](#)

LongitudinalProfile

Purpose To specify the longitudinal profile of a maneuver's activity.

Where to go from here Information in this section

Class Description (LongitudinalProfile (Maneuver)).....	218
To describe the class and its attributes.	
Activate.....	219
To get a specific longitudinal type.	

Class Description (LongitudinalProfile (Maneuver))

Syntax `LongitudinalProfile = ManeuverActivity.LongitudinalType`

Purpose To specify the longitudinal profile of a maneuver's activity.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Velocity, Stop, PedalStimulus, Standstill, or LateralAcceleration ¹⁾	To get the active longitudinal type object.
AvailableElements	String[]	To get the names of all available longitudinal types.

¹⁾ Refer to [Velocity](#) on page 382, [Stop](#) on page 373, [PedalStimulus](#) on page 349, [Standstill](#) on page 371, or [LateralAcceleration](#) on page 338.

Methods The class contains the following methods:

Method	Purpose
Activate	To get a specific longitudinal type. Refer to Activate on page 219.

Related topics

References

ManeuverActivity.....	223
---------------------------------------	---------------------

Activate

Class LongitudinalType

Syntax `LongitudinalType = LongitudinalProfile.Activate(Type)`

Purpose To get a specific longitudinal type.

Parameters The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific longitudinal type. The value can be specified as string (name) or integer (constant type).

Return value The method returns an object of the following type:

Type	Description
Velocity, Stop, PedalStimulus, Standstill, or LateralAcceleration ¹⁾	A longitudinal type object according to the given parameter.

¹⁾ Refer to [Velocity](#) on page 382, [Stop](#) on page 373, [PedalStimulus](#) on page 349, [Standstill](#) on page 371, or [LateralAcceleration](#) on page 338.

Related topics

References

[Class Description \(LongitudinalProfile \(Maneuver\)\)](#)..... 218

Maneuver

Purpose To get a maneuver.

Class Description (Maneuver)

Syntax `Maneuver = Maneuvers.Item(Index)`

```
Maneuver = Maneuvers.Add()
```

Purpose

To get a maneuver.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
CompleteName	String	To get the complete name.
Comment	String	To get/set a comment of the maneuver.
Index	Integer	To get/set the index of the maneuver.
Name	String	To get/set the maneuver name.
Sequences	ManeuverSequences ¹⁾	To get the sequences of the maneuver.
TrafficDriverType	TrafficDriverTypes ²⁾	To get the traffic driver.
Valid	Boolean	To get the information whether the maneuver is valid.

¹⁾ Refer to [ManeuverSequences](#) on page 231.

²⁾ Refer to [TrafficDriverTypes](#) on page 314.

Methods

—

Related topics**References**

[Maneuvers.....](#) 220

Maneuvers

Purpose

To manage the maneuvers of a scenario.

Where to go from here**Information in this section**

[Class Description \(Maneuvers\).....](#) 221
To describe the class and its attributes.

[Add.....](#) 221
To add a maneuver.

Item.....	222
To get a specific maneuver.	
Remove.....	223
To remove a maneuver.	

Class Description (Maneuvers)

Syntax

```
Maneuvers = ActiveTrafficScenario.Maneuver
```

Purpose

To manage the maneuvers of a scenario.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of maneuvers.

Methods

The class contains the following methods:

Method	Purpose
Add	To add a maneuver. Refer to Add on page 221.
Item	To get a specific maneuver. Refer to Item on page 222.
Remove	To remove a maneuver. Refer to Remove on page 223.

Related topics

References

ActiveTrafficScenario.....	198
--	---------------------

Add

Class

Maneuvers

Syntax

```
Maneuver = Maneuvers.Add()
```

Purpose To add a maneuver.

Parameters –

Return value The method returns the following parameter:

Type	Description
Maneuver ¹⁾	The added maneuver.

¹⁾ Refer to [Maneuver](#) on page 219.

Related topics

References

[Class Description \(Maneuvers\)..... 221](#)

Item

Class Maneuvers

Syntax `Maneuver = Maneuvers.Item(Index)`

Purpose To get a specific maneuver.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	Object	The index of the maneuver.

Return value The method returns the following parameter:

Type	Description
Maneuver ¹⁾	The specific maneuver.

¹⁾ Refer to [Maneuver](#) on page 219.

Related topics**References**

[Class Description \(Maneuvers\)..... 221](#)

Remove

Class

Maneuvers

Syntax

```
RetVal = Maneuvers.Remove(Index)
```

Purpose

To remove a maneuver.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	object	The maneuver to be removed.

Return value

The method returns the following parameter:

Type	Description
Boolean	True if successful.

Related topics**References**

[Class Description \(Maneuvers\)..... 221](#)

ManeuverActivity

Purpose

To get a maneuver activity.

Class Description (ManeuverActivity)

Syntax

```
ManeuverActivity = ManeuverSegment.Activity
```

Purpose

To get a maneuver activity.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
LongitudinalType	LongitudinalProfile ¹⁾	To get the longitudinal type.
LateralType	LateralProfile ²⁾	To get the lateral type.
ClutchAndGearType	GearProfile ³⁾	To get the clutch and gear type.
SelectorLeverType	SelectorProfile ⁴⁾	To get the selector lever type.

¹⁾ Refer to [LongitudinalProfile](#) on page 258.

²⁾ Refer to [LateralProfile](#) on page 256.

³⁾ Refer to [GearProfile](#) on page 214.

⁴⁾ Refer to [SelectorProfile](#) on page 236.

Methods

—

Related topics

References

[ManeuverSegment..... 224](#)

ManeuverSegment

Purpose

To get a maneuver segment.

Class Description (ManeuverSegment)

Syntax

```
ManeuverSegment = ManeuverSegments.Item(ID)
ManeuverSegment = ManeuverSegments.Append()
ManeuverSegment = ManeuverSegments.Insert(ID, Type)
```


Purpose To get a maneuver segment.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Activity	ManeuverActivity ¹⁾	To get the activity of the segment.
Comment	String	To get/set a comment for the segment.
Name	String	To get/set a name for the segment.
Successor	ManeuverSegment ²⁾	To get the succeeding segment.
Transition	ManeuverTransition ³⁾	To get the transition of the segment.
UserOutputs	UserOutputs ⁴⁾	To get the user signals.
Valid	Boolean	To get the information whether the segment is valid.

¹⁾ Refer to [ManeuverActivity](#) on page 223.

²⁾ Refer to [ManeuverSegment](#) on page 224.

³⁾ Refer to [ManeuverTransition](#) on page 234.

⁴⁾ Refer to [UserOutputs](#) on page 262.

Methods —

Related topics

References

[ManeuverSegments.....](#) 225

ManeuverSegments

Purpose To get the maneuver segments.

Where to go from here

Information in this section

Class Description (ManeuverSegments).....	226
To describe the class and its attributes.	
Append.....	227
To append a new segment at the end of the segments.	
Insert.....	227
To insert a new segment after or before a specific segment.	

Item.....	228
To get a specific segment.	
Remove.....	229
To remove a segment.	

Class Description (ManeuverSegments)

Syntax

```
ManeuverSegments = ManeuverSequence.Segments
```

Purpose

To get the maneuver segments.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of segments.

Methods

The class contains the following methods:

Method	Purpose
Append	To append a new segment at the end of the segments. Refer to Append on page 227.
Insert	To insert a new segment after or before a specific segment. Refer to Insert on page 227.
Item	To get a specific segment. Refer to Item on page 228.
Remove	To remove a segment. Refer to Remove on page 229.

Related topics

References

ManeuverSequence.....	229
---------------------------------------	---------------------

Append

Class ManeuverSegments

Syntax `ManeuverSegment = ManeuverSegments.Append()`

Purpose To append a new segment at the end of the segments.

Parameters –

Return value The method returns the following parameter:

Type	Description
ManeuverSegment ¹⁾	The new segment.

¹⁾ Refer to [ManeuverSegment](#) on page 224.

Related topics

References

[Class Description \(ManeuverSegments\)..... 226](#)

Insert

Class ManeuverSegments

Syntax `ManeuverSegment = ManeuverSegments.Insert(ID, Type)`

Purpose To insert a new segment after or before a specific segment.

Parameters

The method uses the following parameters:

Parameter	Type	Description
ID	object	The identifier of the segment after or before which the new segment is inserted.
Type	InsertTypes ¹⁾	The insertion type.

¹⁾ Refer to [InsertTypes](#) on page 387.

Return value

The method returns the following parameter:

Type	Description
ManeuverSegment ¹⁾	The new segment.

¹⁾ Refer to [ManeuverSegment](#) on page 224.

Related topics**References**

[Class Description \(ManeuverSegments\)..... 226](#)

Item

Class

ManeuverSegments

Syntax

```
ManeuverSegment = ManeuverSegments.Insert(ID)
```

Purpose

To get a specific segment.

Parameters

The method uses the following parameters:

Parameter	Type	Description
ID	Object	The identifier of the specific segment.

Return value

The method returns the following parameter:

Type	Description
ManeuverSegment ¹⁾	The specific segment.

¹⁾ Refer to [ManeuverSegment](#) on page 224.

Related topics**References**

[Class Description \(ManeuverSegments\)..... 226](#)

Remove

Class

ManeuverSegments

Syntax

```
RetVal = ManeuverSegments.Remove(ID)
```

Purpose

To remove a segment.

Parameters

The method uses the following parameters:

Parameter	Type	Description
ID	object	The identifier of the segment to be removed.

Return value

The method returns the following parameter:

Type	Description
Boolean	True if successful.

Related topics**References**

[Class Description \(ManeuverSegments\)..... 226](#)

ManeuverSequence

Purpose

To get a sequence of a maneuver.

Class Description (ManeuverSequence)

Syntax

```
Sequence = ManeuverSequences.Item(Index)
Sequence = ManeuverSequences.Add()
```

Purpose

To get a sequence of a maneuver.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
AdditionalLateralOffset	Double	To get/set an additional lateral offset.
InitialHeight	Double	To get/set the initial height.
InitialHeightMode	HeightModeTypes ¹⁾	To get/set the mode fore specifying the initial height.
InitialLaneIndex	Integer	To get/set the initial lane index.
InitialLongitudinalVelocity	Double	To get/set the initial velocity in longitudinal direction.
Route	RouteSelection ²⁾	To get the route.
Segments	ManeuverSegments ³⁾	To get the segments of the maneuver sequence
StartCondition	ManeuverTransition ⁴⁾	To get the start condition.
StartPosition	Double	To get/set the start position.
VehicleOrientation	Double	To get/set the vehicle orientation.

¹⁾ Refer to [HeightModeTypes](#) on page 387.

²⁾ Refer to [RouteSelection](#) on page 358.

³⁾ Refer to [ManeuverSegments](#) on page 225.

⁴⁾ Refer to [ManeuverTransition](#) on page 234.

Methods

—

Related topics

References

[ManeuverSequences.....](#) 231

ManeuverSequences

Purpose To get all the sequences of a maneuver.

Where to go from here

Information in this section

Class Description (ManeuverSequences)	231
To describe the class and its attributes.	
Add	232
To add a sequence.	
Item	232
To get a specific sequence.	
Remove	233
To remove a sequence.	

Class Description (ManeuverSequences)

Syntax

```
ManeuverSequences = Maneuver.Sequences
```

Purpose To get all the sequences of a maneuver.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of sequences.

Methods

The class contains the following methods:

Method	Purpose
Add	To add a sequence. Refer to Add on page 232.
Item	To get a specific sequence. Refer to Item on page 232.
Remove	To remove a sequence. Refer to Remove on page 233.

Related topics**References**

[Maneuver.....](#) 219

Add

Class

ManeuverSequences

Syntax

```
Sequence = ManeuverSequences.Add()
```

Purpose

To add a sequence.

Parameters

—

Return value

The method returns the following parameter:

Type	Description
ManeuverSequence ¹⁾	The added sequence.

¹⁾ Refer to [ManeuverSequence](#) on page 229.

Related topics**References**

[Class Description \(ManeuverSequences\).....](#) 231

Item

Class

ManeuverSequences

Syntax

```
Sequence = ManeuverSequences.Item(Index)
```


Purpose To get a specific sequence.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	Object	The index of the sequence.

Return value The method returns the following parameter:

Type	Description
ManeuverSequence ¹⁾	The specific sequence.

¹⁾ Refer to [ManeuverSequence](#) on page 229.

Related topics

References

[Class Description \(ManeuverSequences\)..... 231](#)

Remove

Class ManeuverSequences

Syntax `RetVal = ManeuverSequences.Remove(Index)`

Purpose To remove a sequence.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	object	The index of the sequences to be removed.

Return value The method returns the following parameter:

Type	Description
Boolean	True if successful

Related topics**References**[Class Description \(ManeuverSequences\)..... 231](#)

ManeuverTransition

Purpose

To get the transition of a maneuver.

Class Description (ManeuverTransition)

Syntax`ManeuverTransition = ManeuverSegment.Transition`**Purpose**

To get the transition of a maneuver.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get/set the transition name.
Comment	String	To get/set the comment/description of the transition.
Delay	Double	To get/set the delay value.
Valid	Boolean	To get the information whether the transition is valid.
Conditions	TransitionsConditions ¹⁾	To get the condition types.

¹⁾ Refer to [TransitionsConditions](#) on page 293.**Methods**

—

Related topics**References**[ManeuverSegment..... 224](#)

OpenClutch

Purpose To specify the OpenClutch type.

Class Description (OpenClutch)

Syntax `OpenClutch = GearProfile.ActiveElement`

Purpose To specify the OpenClutch type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the element.
Type	ClutchAndGearTypes ¹⁾	To get the type of the element.

¹⁾ Refer to [ClutchAndGearTypes](#) on page 385.

Methods —

Related topics

References

[GearProfile.....](#) 214

ReferenceGear

Purpose To specify a ReferenceGear type.

Class Description (ReferenceGear)

Syntax `ReferenceGear = GearProfile.ActiveElement`

Purpose To specify a ReferenceGear type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the element.
Type	ClutchAndGearTypes ¹⁾	To get the type of the element.
GearSource	Constant, Table, Manual, or Extern ²⁾	To get the source type of the gear source.

¹⁾ Refer to [ClutchAndGearTypes](#) on page 385.

²⁾ Refer to [Constant](#) on page 362, [Table](#) on page 370, [Manual](#) on page 365, or [Extern](#) on page 362.

Methods —

Related topics

References

[GearProfile](#)..... 214

SelectorProfile

Purpose To get the selector lever type.

Where to go from here

Information in this section

[Class Description \(SelectorProfile\)](#)..... 237

To describe the class and its attributes.

[Activate](#)..... 237

To activate an element for the selector profile.

Class Description (SelectorProfile)

Syntax

```
SelectorProfile = ManeuverActivity.SelectorLeverType
```

Purpose

To get the selector lever type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Driver or Stimulus ¹⁾	To get the active element.
AvailableElements	String[]	To get the names of all the available elements.

¹⁾ Refer to [Driver](#) on page 212 or [Stimulus](#) on page 238.

Methods

The class contains the following methods:

Method	Purpose
Activate	To activate an element. Refer to Activate on page 237.

Related topics

References

[ManeuverActivity..... 223](#)

Activate

Class

SelectorProfile

Syntax

```
ActivatedElement = SelectorProfile.Activate(Type)
```

Purpose

To activate an element for the selector profile.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Type	Driver or Stimulus ¹⁾	The type to be activated.

¹⁾ Refer to [Driver](#) on page 212 or [Stimulus](#) on page 238.

Return value

The method returns the following parameter:

Type	Description
Driver or Stimulus ¹⁾	The activated element.

¹⁾ Refer to [Driver](#) on page 212 or [Stimulus](#) on page 238.

Related topics**References**

[Class Description \(SelectorProfile\)..... 237](#)

Stimulus

Purpose

To specify a stimulus type.

Class Description (Stimulus)

Syntax

```
Stimulus = GearProfile.ActiveElement
```

Purpose

To specify a stimulus type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the element.
Type	ClutchAndGearTypes ¹⁾	To get the clutch and gear type.

Attributes	Type	Purpose
GearSource	Constant, Table, Manual, or Extern ²⁾	To get the source type of the gear source.
ClutchSource	Constant, Table, Manual, Extern, or Ramp ³⁾	To get the source type of the clutch source.

- ¹⁾ Refer to [ClutchAndGearTypes](#) on page 385.
- ²⁾ Refer to [Constant](#) on page 362, [Table](#) on page 370, [Manual](#) on page 365, or [Extern](#) on page 362.
- ³⁾ Refer to [Constant](#) on page 362, [Table](#) on page 370, [Manual](#) on page 365, [Extern](#) on page 362, or [Ramp](#) on page 366.

Methods —

Related topics

References

GearProfile.....	214
----------------------------------	-----

Classes for Configuring Fellows

Where to go from here

Information in this section

Fellow	241
To get information about a fellow.	
FellowActivity	242
To get information about the activity of a fellow.	
Fellows	242
Containing all fellows of the active scenario.	
FellowSegment	246
Containing the information of a single segment of a fellow in a scenario (including activity, transition, and fellow user signal).	
FellowSegments	246
This is the collection of all segments of a fellow definition.	
FellowSequence	251
To get the sequence of a fellow.	
FellowSequences	251
This is the collection of all sequences of a fellow definition.	
FellowTransition	255
To get the transition of a fellow.	
LateralProfile	256
To get lateral types of a fellow definition.	
LongitudinalProfile	258
To get longitudinal types of an activity.	
TrafficObjectTypes	260
To assign or get information about the assigned traffic object type and get a collection of all the traffic objects.	
TrafficObjectType	261
To get information about a traffic object type.	
UserOutputs	262
To get information about all available fellow and maneuver user signals of an activity.	
UserOutput	264
To get information about a fellow user signal.	

Information in other sections

Overview of the Classes for Creating Scenarios	187
The overview gives you a quick overview of the classes for the Scenario Editor.	

[Automating Scenarios in Python.....](#) 180
You can change the parameters of the scenario.

Fellow

Purpose To get information about a fellow.

Class Description (Fellow)

Syntax `Fellow = Fellows.Add()`

Purpose To get information about a fellow.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get/set the fellow name.
Comment	String	To get/set the comment/description of the fellow.
CompleteName	String	To get the complete fellow name, for example, 'F7: Blue Roadster'.
Index	Integer	To get the matrix index in the model of the fellow.
Sequences	FellowSequences ¹⁾	To get the FellowSequences object.
TrafficDriverType	TrafficDriverTypes ²⁾	To get the traffic driver.
TrafficObjectType	TrafficObjectTypes ³⁾	To get the assigned traffic object type of this fellow.

¹⁾ Refer to [FellowSequences](#) on page 251.

²⁾ Refer to [TrafficDriverTypes](#) on page 314.

³⁾ Refer to [TrafficObjectTypes](#) on page 260.

Related topics

References

[Fellows.....](#) 242

FellowActivity

Purpose To get information about the activity of a fellow.

Class Description (FellowActivity)

Syntax No direct creation

Purpose To get information about the activity of a fellow.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
LongitudinalType	LongitudinalProfile ¹⁾	To get the longitudinal type definition of the activity.
LateralType	LateralProfile ²⁾	To get the lateral type definition of the activity.

¹⁾ Refer to [LongitudinalProfile](#) on page 258.

²⁾ Refer to [LateralProfile](#) on page 256.

Fellows

Purpose The class contains all fellows of the active scenario.

Where to go from here

Information in this section

Class Description (Fellows)	243
To describe the class and its attributes.	
Add	243
To add a fellow to the fellows collection.	
Item	244
To get a specific fellow from the fellows collection.	
Remove	245
To remove a specific fellow from the fellows collection.	

Class Description (Fellows)

Syntax Is not created directly

Purpose The class contains all information and parameters about the fellows of the active scenario.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of fellows in the scenario.

Methods The class contains the following methods:

Method	Purpose
Add	To add a fellow to the fellows collection. Refer to Add on page 243.
Item	To get a specific fellow from the fellows collection. Refer to Item on page 244.
Remove	To remove a specific fellow from the fellows collection. Refer to Remove on page 245.

Related topics

HowTos

[How to Create and Specify a Fellow.....](#) 62

References

[Fellow Properties.....](#) 126

Add

Class Fellows

Syntax `Fellow = Fellows.Add()`

Purpose To add a fellow to the fellows collection.

Parameters

—

Return value

The method returns an object of the following type:

Type	Description
Fellow ¹⁾	The new fellow object.

¹⁾ Refer to [Fellow](#) on page 241.

Related topics**References**

[Class Description \(Fellows\)..... 243](#)

Item

Class

Fellows

Syntax

```
Fellow = Fellows.Item(Index)
```

Purpose

To get a specific fellow from the fellows collection.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	Variant	<p>The index of the specific fellow. Strings (fellow name) and integers (fellow index) are supported.</p> <div> Note <p>The index can differ from the index that is displayed in ModelDesk if the simulation ID has been modified. Use the simulation ID to clearly identify a fellow.</p> </div>

Return value

The method returns an object of the following type:

Type	Description
Fellow ¹⁾	The specific fellow object.

¹⁾ Refer to [Fellow](#) on page 241.

Related topics**References**

[Class Description \(Fellows\)..... 243](#)

Remove

Class

Fellows

Syntax

```
Fellow = Fellows.Remove(Index)
```

Purpose

To remove a specific fellow from the fellows collection.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	Variant	The index of the specific fellow. Strings (fellow name), integers (fellow index) and the fellow itself are supported.

Return value

The method returns an object of the following type:

Type	Description
Boolean	True if successful

Related topics**References**

[Class Description \(Fellows\)..... 243](#)

FellowSegment

Purpose The class contains the information of a single segment of a fellow in a scenario (including activity, transition, and fellow user signal).

Class Description (FellowSegment)

Syntax No direct creation

Purpose To get information of a single segment of a fellow in a scenario (including activity, transition, and fellow user signal).

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Activity	FellowActivity ¹⁾	To get the FellowActivity object.
Transition	FellowTransition ²⁾	To get the FellowTransition object.
UserOutputs	UserOutputs ³⁾	To get the UserOutputs object.
Comment	String	To get/set the comment of the segment.
Name	String	To get/set the segment name.
Valid	Boolean	To get the valid state of the segment.

¹⁾ Refer to [FellowActivity](#) on page 242.

²⁾ Refer to [FellowTransition](#) on page 255.

³⁾ Refer to [UserOutputs](#) on page 262.

FellowSegments

Purpose This is the collection of all segments of a fellow definition.

Where to go from here

Information in this section

[Class Description \(FellowSegments\)](#)..... 247
To describe the class and its attributes.

Append.....	248
To append a new FellowSegment object to the FellowSegments collection.	
Insert.....	248
To insert a new FellowSegment object to the FellowSegments collection.	
Item.....	249
To get a FellowSegment object specified by the given index.	
Remove.....	250
To remove a specific FellowSegment object from the FellowSegments collection.	

Class Description (FellowSegments)

Syntax

```
FellowSegments = FellowSequence.Segments
```

Purpose

To handle a collection of segments of a fellow.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of FellowSegment objects.

Methods

The class contains the following methods:

Method	Purpose
Append	To append a new FellowSegment object to the FellowSegments collection. Refer to Append on page 248.
Insert	To insert a new FellowSegment object to the FellowSegments collection. Refer to Insert on page 248.
Item	To get a specific FellowSegment object from the collection. Refer to Item on page 249.
Remove	To remove a specific FellowSegment object from the FellowSegments collection. Refer to Remove on page 250.

Related topics**References**

[FellowSequence](#)..... 251

Append

Class

FellowSegments

Syntax

```
FellowSegment = FellowSegments.Append()
```

Purpose

To append a new FellowSegment object to the FellowSegments collection.

Parameters

—

Return value

The method returns an object of the following type:

Type	Description
FellowSegment ¹⁾	The new fellow segment.

¹⁾ Refer to [FellowSegment](#) on page 246.

Related topics**References**

[Class Description \(FellowSegments\)](#)..... 247

Insert

Class

FellowSegments

Syntax

```
FellowSegment = FellowSegments.Insert(Index, Type)
```


Purpose To insert a new FellowSegment object to the FellowSegments collection.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	Variant	Index of a specific FellowSegment object to which the new FellowSegment object is inserted before or after. The value can be specified as object (FellowSegment) or integer (index).
Type	InsertTypes ¹⁾	Specifies whether the new FellowSegment object is inserted before or after the specified FellowSegment object.

¹⁾ Refer to [InsertTypes](#) on page 387.

Return value The method returns an object of the following type:

Type	Description
FellowSegment ¹⁾	The new FellowSegment object.

¹⁾ Refer to [FellowSegment](#) on page 246.

Related topics

References

[Class Description \(FellowSegments\)..... 247](#)

Item

Class FellowSegments

Syntax `FellowSegment = FellowSegments.Item(Index)`

Purpose To get a specific FellowSegment object from the FellowSegments collection.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	Variant	Index of the specific fellow segment. The value can be specified as string (name) or integer (index).

Return value

The method returns an object of the following type:

Type	Description
FellowSegment ¹⁾	The specified segment.

¹⁾ Refer to [FellowSegment](#) on page 246.

Related topics**References**

[Class Description \(FellowSegments\)..... 247](#)

Remove

Class

FellowSegments

Syntax

```
Result = FellowSegments.Remove(Index)
```

Purpose

To remove a specific FellowSegment object from the FellowSegments collection.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	Variant	Index of the FellowSegment object to be removed. You can specify the value as string (name) or integer (index).

Return value

The method returns an object of the following type:

Type	Description
Boolean	True if successful.

Related topics**References**

[Class Description \(FellowSegments\)..... 247](#)

FellowSequence

Purpose To get the sequence of a fellow.

Class Description (FellowSequence)

Syntax `FellowSequence = FellowSequences.Item(Index)`

Purpose To get the sequence of a fellow.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Segments	FellowSegments ¹⁾	To get the fellow segments.
StartCondition	FellowTransition ²⁾	To get the start transition.
Route	RouteSelection ³⁾	To get the route for the fellow.

¹⁾ Refer to [FellowSegments](#) on page 246.

²⁾ Refer to [FellowTransition](#) on page 255.

³⁾ Refer to [RouteSelection](#) on page 358.

Related topics

References

[FellowSequences..... 251](#)

FellowSequences

Purpose This is the collection of all sequences of a fellow definition.

Where to go from here

Information in this section

[Class Description \(FellowSequences \(Collection\)\)..... 252](#)
To describe the class and its attributes.

Add.....	253
To add a new FellowSequence object to the collection.	
Item.....	253
To get a FellowSequence object from the FellowSequences collection.	
Remove.....	254
To remove a specific FellowSequence object from the collection.	

Class Description (FellowSequences (Collection))

Syntax

```
FellowSequences = Fellow.Sequences
```

Purpose

To handle a collection of sequences of a fellow.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of FellowSequence objects.

Methods

The class contains the following methods:

Method	Purpose
Add	To add a new FellowSequence object to the collection. Refer to Add on page 253.
Item	To get a specific FellowSequence object from the collection. Refer to Item on page 253.
Remove	To remove a specific FellowSequence object from the collection. Refer to Remove on page 254.

Related topics

References

Fellow.....	241
-----------------------------	---------------------

Add

Class FellowSequences

Syntax `Result = FellowSegments.Remove(Index)`

Purpose To add a new FellowSequence object to the collection.

Parameters –

Return value The method returns an object of the following type:

Type	Description
FellowSequence ¹⁾	The new FellowSequence object.

¹⁾ Refer to [FellowSequence](#) on page 251.

Related topics

References

[FellowSequences..... 251](#)

Item

Class FellowSequences

Syntax `FellowSequence = FellowSequences.Item(Index)`

Purpose To get a FellowSequence object from the FellowSequences collection.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	Variant	Index of a specific FellowSequence object. The value can be specified as string (name) or integer (index).

Return value

The method returns an object of the following type:

Type	Description
FellowSequence ¹⁾	The specified FellowSequence object.

¹⁾ Refer to [FellowSequence](#) on page 251.

Related topics**References**

[FellowSequences..... 251](#)

Remove

Class

FellowSequences

Syntax

```
Result = FellowSequences.Remove(Index)
```

Purpose

To remove a specific FellowSequence object from the collection.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	Variant	Index of the FellowSequence object to be removed. The value can be specified as FellowSequence object or integer (index).

Return value

The method returns an object of the following type:

Type	Description
Bool	True if successful

Related topics**References**

[FellowSequences..... 251](#)

FellowTransition

Purpose To get the transition of a fellow.

Class Description (FellowTransition)

Syntax

```
FellowTransition = FellowSequence.StartCondition
FellowTransition = FellowSegment.Transition
```

Purpose To get the transition of a fellow.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get/set the transition name.
Comment	String	To get/set the comment/description of the transition.
Delay	Double	To get/set the delay value.
Valid	Boolean	To get the information whether the transition is valid.
Conditions	TransitionsConditions ¹⁾	To get the condition types.

¹⁾ Refer to [TransitionsConditions](#) on page 293.

Related topics

References

FellowSegment	246
FellowSequence	251

LateralProfile

Purpose The class contains lateral types of a fellow definition.

Where to go from here

Information in this section

[Class Description \(LateralProfile\).....256](#)

To describe the class and its attributes.

[Activate.....257](#)

To get a specific lateral type.

Class Description (LateralProfile)

Syntax

```
LateralProfile = FellowActivity.LateralType
```

Purpose To get lateral types of an activity.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	NotUsed, Continue, LateralDeviation, or LaneSelection ¹⁾	To get the active lateral type object.
AvailableElements	Strings (collection)	To get the names of all available lateral types.

¹⁾ Refer to [NotUsed](#) on page 348, [Continue](#) on page 291, [LateralDeviation](#) on page 339, or [LaneSelection](#) on page 341.

Methods

The class contains the following methods:

Method	Purpose
Activate	To activate a lateral type. Refer to Activate on page 257.

Related topics**References**

[FellowActivity](#)..... 242

Activate

Class

LateralProfile

Syntax

```
LateralType = LateralProfileType.Activate(Type)
```

Purpose

To get a specific lateral type.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific lateral type. The value can be specified as string (name) or integer (constant type).

Return value

The method returns an object of the following type:

Type	Description
NotUsed, Continue, LateralDeviation, or LaneSelection ¹⁾	A lateral type object according to the given parameter.

¹⁾ Refer to [NotUsed](#) on page 348, [Continue](#) on page 291, [LateralDeviation](#) on page 339, or [LaneSelection](#) on page 341.

Related topics**References**

[Class Description \(LateralProfile\)](#)..... 256

LongitudinalProfile

Purpose The class contains longitudinal types of an activity.

Where to go from here

Information in this section

Class Description (LongitudinalProfile).....	258
To describe the class and its attributes.	
Activate.....	259
To get a specific longitudinal type.	

Class Description (LongitudinalProfile)

Syntax

```
LongitudinalProfile = FellowActivity.LongitudinalType
```

Purpose

To get the longitudinal type of a fellow's activity.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	NotUsed, Continue, Acceleration, DistanceMeter, DistanceTime, Velocity, or StopWithin ¹⁾	To get the active longitudinal type object.
AvailableElements	Strings (collection)	To get the names of all available longitudinal types.

¹⁾ Refer to [NotUsed](#) on page 348, [Continue](#) on page 292, [Acceleration](#) on page 323, [DistanceMeter](#) on page 332, [DistanceTime](#) on page 334, [Position](#) on page 350, [Velocity](#) on page 382, or [StopWithin](#) on page 374.

Methods

The class contains the following methods:

Method	Purpose
Activate	To activate a longitudinal type. Refer to Activate on page 259.

Related topics**References**

[FellowActivity](#)..... 242

Activate

Class

LongitudinalType

Syntax

```
LongitudinalType = LongitudinalProfile.Activate(Type)
```

Purpose

To get a specific longitudinal type.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific longitudinal type. The value can be specified as string (name) or integer (constant type).

Return value

The method returns an object of the following type:

Type	Description
NotUsed, Continue, Acceleration, DistanceMeter, DistanceTime, Position, Velocity, or StopWithin ¹⁾	A longitudinal type object according to the given parameter.

¹⁾ Refer to [NotUsed](#) on page 348, [Continue](#) on page 292, [Acceleration](#) on page 323, [DistanceMeter](#) on page 332, [DistanceTime](#) on page 334, [Position](#) on page 350, [Velocity](#) on page 382, or [StopWithin](#) on page 374.

Related topics**References**

[Class Description \(LongitudinalProfile\)](#)..... 258

TrafficObjectTypes

Purpose To assign or get information about the assigned traffic object type and get a collection of all the traffic objects.

Where to go from here **Information in this section**

[Class Description \(TrafficObjectTypes\)](#)..... 260
To describe the class and its attributes.

[Activate](#)..... 261
To activate a traffic object type.

Class Description (TrafficObjectTypes)

Syntax `TrafficObjectTypes = Fellow.TrafficObjectTypes`

Purpose To assign or get information about the assigned traffic object type and get a collection of all the traffic objects.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	TrafficObjectType ¹⁾	To get the active TrafficObjectType object.
AvailableElements	Strings (collection)	To get the names of all available fellow types.

¹⁾ Refer to [TrafficObjectType](#) on page 261.

Methods The class contains the following methods:

Method	Purpose
Activate	To activate a traffic object type. Refer to Activate on page 261.

Related topics

References

[Fellow](#)..... 241

Activate

Class TrafficObjectTypes

Syntax `TrafficObjectType = TrafficObjectTypes.Activate(Item)`

Purpose To activate a traffic object type.

Parameters The method uses the following parameters:

Parameter	Type	Description
Item	Variant	The identifier of the traffic object type. Only traffic object type names with or without file name extension (string) are supported.

Return value The method returns an object of the following type:

Type	Description
TrafficObjectType ¹⁾	The activated TrafficObjectType object.

¹⁾ Refer to [TrafficObjectType](#) on page 261.

Related topics

References

[Class Description \(TrafficObjectTypes\)](#)..... 260

TrafficObjectType

Purpose To get information about a traffic object type.

Class Description (TrafficObjectType)

Syntax No direct creation

Purpose To get information about a traffic object type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Path	String	To get the path including the file name and extension of the traffic object type.
Comment	String	To get the comment/description of the traffic object type.

UserOutputs

Purpose To get information about all available fellow and maneuver user signals of an activity.

Where to go from here Information in this section

Class Description (UserOutputs (Collection)).....	262
To describe the class and its attributes.	
Item.....	263
To get a specific user signal from the collection.	

Class Description (UserOutputs (Collection))

Syntax

```
UserOutputs = FellowSegment.UserOutputs
UserOutputs = ManeuverSegment.UserOutputs
```

Purpose To get information about all available fellow or maneuver user signals of an activity.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of all available user outputs in the segment.

Methods

The class contains the following methods:

Method	Purpose
Item	To get a specific user signal from the collection. Refer to Item on page 263.

Related topics**References**

FellowSegment	246
ManeuverSegment	224

Item

Class

UserOutputs

Syntax

```
UserOutput = UserOutputs.Item(Index)
```

Purpose

To get a specific user signal from the collection.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	Variant	The index of the specific user signal. Integers (index) are supported.

Return value

The method returns an object of the following type:

Type	Description
UserOutput ¹⁾	The user signal.

¹⁾ Refer to [UserOutput](#) on page 264.

Related topics**References**[Class Description \(UserOutputs \(Collection\)\)..... 262](#)

UserOutput

Purpose

To get information about a fellow user signal.

Class Description (UserOutput)

Syntax

```
UserOutput = UserOutputs.Item(Index)
```

Purpose

To get information about a user signal.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the alias of the user signal.
Value	UserSignalValue ¹⁾	To get the value of the user signal if isUsed is True.
IsUsed	Boolean	To get/set the state whether the user signal is used.
Valid	Boolean	To get the valid state.
Comment	String	To get the comment/description of the user signal.

¹⁾ Refer to [UserSignalValue](#) on page 280.

Related topics**References**[UserOutputs..... 262](#)

Classes for Configuring Global User Signals

Where to go from here

Information in this section

GlobalUserSignals.....	266
To administrate the global user signals.	
GlobalUserSignal.....	269
To access a global user signal.	
UserSignalSegments.....	270
To administrate the segments of a global user signal.	
UserSignalSegment.....	274
To access a segment of a global user signal.	
UserSignalSequences.....	275
To administrate the sequences of a global user signal.	
UserSignalSequence.....	278
To access a sequence of a global user signal.	
UserSignalTransition.....	279
To get information on the transition of a segment.	
UserSignalValue.....	280
To get the value of the global user signal.	

Information in other sections

Overview of the Classes for Creating Scenarios.....	187
The overview gives you a quick overview of the classes for the Scenario Editor.	
Automating Scenarios in Python.....	180
You can change the parameters of the scenario.	

GlobalUserSignals

Purpose To administrate the global user signals.

Where to go from here

Information in this section

Class Description (GlobalUserSignals)	266
To describe the class and its attributes.	
Add	267
To add a GlobalUserSignal object to the collection.	
Item	267
To get a specific GlobalUserSignal object.	
Remove	268
To remove a specific GlobalUserSignal object.	

Class Description (GlobalUserSignals)

Syntax `GlobalUserSignals = ActiveTrafficScenario.GlobalUserSignals`

Purpose To administrate the global user signals.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of GlobalUserSignal objects.

Methods The class contains the following methods:

Method	Purpose
Add	To add a GlobalUserSignal object to the collection. Refer to Add on page 267.
Item	To get a specific GlobalUserSignal object. Refer to Item on page 267.
Remove	To remove a specific GlobalUserSignal object. Refer to Remove on page 268.

Related topics**References**

[ActiveTrafficScenario..... 198](#)

Add

Class

GlobalUserSignals

Syntax

```
GlobalUserSignal = GlobalUserSignals.Add()
```

Purpose

To add a GlobalUserSignal object to the collection.

Parameters

—

Return value

The method returns an object of the following type:

Type	Description
GlobalUserSignal ¹⁾	The new GlobalUserSignal object.

¹⁾ Refer to [GlobalUserSignal](#) on page 269.

Related topics**References**

[Class Description \(GlobalUserSignals\)..... 266](#)

Item

Class

GlobalUserSignals

Syntax

```
GlobalUserSignal = GlobalUserSignals.Item(Index)
```

Purpose To get a specific GlobalUserSignal object.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	Variant	The index of the specific GlobalUserSignal object. The value can be specified as string (name) or integer (index).

Return value The method returns an object of the following type:

Type	Description
GlobalUserSignal ¹⁾	The specific GlobalUserSignal object.

¹⁾ Refer to [GlobalUserSignal](#) on page 269.

Related topics

References

[Class Description \(GlobalUserSignals\)..... 266](#)

Remove

Class GlobalUserSignals

Syntax `Result = GlobalUserSignals.Remove(Index)`

Purpose To remove a specific GlobalUserSignal object.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	Variant	The index of the specific GlobalUserSignal object. The value can be specified as string (name), integer (index), or GlobalUserSignal object.

Return value

The method returns an object of the following type:

Type	Description
Boolean	True if successful

Related topics**References**

[Class Description \(GlobalUserSignals\).....](#) 266

GlobalUserSignal

Purpose

To access a global user signal.

Class Description (GlobalUserSignal)

Syntax

```
GlobalUserSignal = GlobalUserSignals.Item(Index)
```

Purpose

To access a global user signal.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get/set the name of the global user signal.
Comment	String	To get/set a comment for the global user signal.
CompleteName	String	To get the complete name such as "U2: My UserSignal", of the global user signal.
Valid	Boolean	To get the valid state.
Index	Integer	To get the simulation ID.
Sequences	UserSignalSequences ¹⁾	To get the UserSignalSequences object.

¹⁾ Refer to [UserSignalSequences](#) on page 275.

Related topics

References

[GlobalUserSignals.....](#) 266

UserSignalSegments

Purpose

To administrate the segments of a global user signal.

Where to go from here

Information in this section

Class Description (UserSignalSegments).....	270
To describe the class and its attributes.	
Append.....	271
To append a segment to the collection.	
Insert.....	272
To insert a segment into the collection.	
Item.....	272
To access a specific segment of the collection.	
Remove.....	273
To remove a specific segment from the collection.	

Class Description (UserSignalSegments)

Syntax

```
UserSignalSegments = UserSignalSequence.Segments
```

Purpose

To administrate the segments of a global user signal.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of segments.

Methods

The class contains the following methods:

Method	Purpose
Append	To append a segment to the collection. Refer to Append on page 271.
Insert	To insert a segment into the collection. Refer to Insert on page 272.
Item	To access a specific segment of the collection. Refer to Item on page 272.
Remove	To remove a specific segment from the collection. Refer to Remove on page 273.

Related topics**References**

[UserSignalSequence](#)..... 278

Append

Class

UserSignalSegments

Syntax

```
UserSignalSegment = UserSignalSegments.Append()
```

Purpose

To append a segment to the collection.

Parameters

—

Return value

The method returns an object of the following type:

Type	Description
UserSignalSegment ¹⁾	The new UserSignalSegment object.

¹⁾ Refer to [UserSignalSegment](#) on page 274.

Related topics**References**

[Class Description \(UserSignalSegments\)](#)..... 270

Insert

Class UserSignalSegments

Syntax `UserSignalSegment = UserSignalSegments.Insert(ID, Type)`

Purpose To insert a segment into the collection.

Parameters The method uses the following parameters:

Parameter	Type	Description
ID	Variant	The index of a specific UserSignalSegment object. The value can be specified as the UserSignalSegment object or integer (constant type).
Type	InsertTypes ¹⁾	Specifies whether the new UserSignalSegment object is inserted before or after the specified UserSignalSegment object specified by the ID parameter.

¹⁾ Refer to [InsertTypes](#) on page 387.

Return value The method returns an object of the following type:

Type	Description
UserSignalSegment ¹⁾	The new UserSignalSegment object.

¹⁾ Refer to [UserSignalSegment](#) on page 274.

Related topics

References

[Class Description \(UserSignalSegments\)..... 270](#)

Item

Class UserSignalSegments

Syntax `UserSignalSegment = UserSignalSegments.Item(ID)`

Purpose To access a specific segment of the collection.

Parameters The method uses the following parameters:

Parameter	Type	Description
ID	Variant	The index of the specific UserSignalSegment object. The value can be specified as string (name) or integer (index).

Return value The method returns an object of the following type:

Type	Description
UserSignalSegment ¹⁾	The specific UserSignalSegment object.

¹⁾ Refer to [UserSignalSegment](#) on page 274.

Related topics

References

[Class Description \(UserSignalSegments\).....](#) 270

Remove

Class UserSignalSegments

Syntax `Result = UserSignalSegments.Remove(ID)`

Purpose To remove a specific segment from the collection.

Parameters The method uses the following parameters:

Parameter	Type	Description
ID	Variant	The index of the specific UserSignalSegment object. The value can be specified as UserSignalSegment object or integer (index).

Return value

The method returns an object of the following type:

Type	Description
Bool	True if successful

Related topics**References**

[Class Description \(UserSignalSegments\).....](#) 270

UserSignalSegment

Purpose

To access a segment of a global user signal.

Class Description (UserSignalSegment)

Syntax

```
UserSignalSegment = UserSignalSegments.Item(Index)
```

Purpose

To access a segment of a global user signal.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get/set the name of the segment.
Comment	String	To get/set the comment of the segment.
Valid	Bool	To get the valid state.
Activity	UserSignalValue ¹⁾	To get the value.
Transition	UserSignalTransition ²⁾	To get UserSignalTransition object.
Successor	UserSignalSegment ³⁾	To get the UserSignalSegment of the successor.

¹⁾ Refer to [UserSignalValue](#) on page 280.

²⁾ Refer to [UserSignalTransition](#) on page 279.

³⁾ Refer to [UserSignalSegment](#) on page 274.

Related topics**References**

[UserSignalSegments..... 270](#)

UserSignalSequences

Purpose

To administrate the sequences of a global user signal.

Where to go from here**Information in this section**

[Class Description \(UserSignalSequences\)..... 275](#)
To describe the class and its attributes.

[Add..... 276](#)
To add a UserSignalSequence object to the collection.

[Item..... 277](#)
To access a specific UserSignalSequence object.

[Remove..... 277](#)
To remove a UserSignalSequence object from the collection.

Class Description (UserSignalSequences)

Syntax

```
UserSignalSequences = GlobalUserSignal.Sequences
```

Purpose

To administrate the sequences of a global user signal.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of sequences of the collection.

Methods

The class contains the following methods:

Method	Purpose
Add	To add a UserSignalSequence object to the collection. Refer to Add on page 276.
Item	To access a specific UserSignalSequence object. Refer to Item on page 277.
Remove	To remove a UserSignalSequence object from the collection. Refer to Remove on page 277.

Related topics**References**

[GlobalUserSignal](#)..... 269

Add

Class

UserSignalSequences

Syntax

```
UserSignalSequence = UserSignalSequences.Add()
```

Purpose

To add a UserSignalSequence object to the collection.

Parameters

—

Return value

The method returns an object of the following type:

Type	Description
UserSignalSequence ¹⁾	The new UserSignalSequence object.

¹⁾ Refer to [UserSignalSequence](#) on page 278.

Related topics**References**

[Class Description \(UserSignalSequences\)](#)..... 275

Item

Class UserSignalSequences

Syntax `UserSignalSequence = UserSignalSequences.Item(Variant Index)`

Purpose To access a specific UserSignalSequence object.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	Variant	The index of the specific UserSignalSequence object. The value can be specified as integer (index).

Return value The method returns an object of the following type:

Type	Description
UserSignalSequence ¹⁾	The specific UserSignalSequence object.

¹⁾ Refer to [UserSignalSequence](#) on page 278.

Related topics

References

[Class Description \(UserSignalSequences\)](#)..... 275

Remove

Class UserSignalSequences

Syntax `Result = UserSignalSequences.Remove(Variant Index)`

Purpose To remove a UserSignalSequence object from the collection.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	Variant	The index of the specific UserSignalSequence object. The value can be specified as UserSignalSequence object or integer (index).

Return value

The method returns an object of the following type:

Type	Description
Bool	True if successful

Related topics**References**

[Class Description \(UserSignalSequences\)..... 275](#)

UserSignalSequence

Purpose

To access a sequence of a global user signal.

Class Description (UserSignalSequence)

Syntax

```
UserSignalSequence = UserSignalSequences.Item(variant Index)
```

Purpose

To access a sequence of a global user signal.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Segments	UserSignalSegments ¹⁾	To get the UserSignalSegments object.
StartCondition	UserSignalTransition ²⁾	To get the UserSignalTransition object that is used as start condition.

¹⁾ Refer to [UserSignalSegments](#) on page 270.

²⁾ Refer to [UserSignalTransition](#) on page 279.

Related topics**References**

[UserSignalSequences.....](#) 275

UserSignalTransition

Purpose

To get information on the transition of a segment.

Class Description (UserSignalTransition)

Syntax

```
UserSignalTransition = UserSignalSegment.Transition
UserSignalTransition = UserSignalSequence.StartCondition
```

Purpose

To get information on the transition of a segment.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get/set the name of the transition.
Comment	String	To get/set the comment of the transition.
Delay	Double	To get/set the time delay.
Conditions	TransitionsConditions ¹⁾	To get the transition conditions.

¹⁾ Refer to [TransitionsConditions](#) on page 293.

Related topics**References**

[UserSignalSegment.....](#) 274
[UserSignalSequence.....](#) 278

UserSignalValue

Purpose To get the value of the global user signal.

Where to go from here

Information in this section

Class Description (UserSignalValue).....	280
To describe the class and its attributes.	
Activate.....	281
To activate a source type for the UserSignalValue object.	

Class Description (UserSignalValue)

Syntax

```
UserSignalValue = UserSignalElement.Activity
```

Purpose

To get the value of the global user signal.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Delay	Double	To get/set a time delay.
ActiveElement	SimpleConstant ¹⁾	To get the active source type object.
AvailableElement	Strings (collection)	To get the names of all available source types.

¹⁾ Refer to [SimpleConstant](#) on page 366.

Methods

The class contains the following methods:

Method	Purpose
Activate	To activate a source type for the UserSignalValue object. Refer to Activate on page 281.

Related topics

References

UserSignalSegment.....	274
--	---------------------

Activate

Class UserSignalValue

Syntax `UserSignalValueType = UserSignalValue.Activate(Variant Type)`

Purpose To activate a source type for the UserSignalValue object.

Parameters The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific UserSignalValue object. The value can be specified as string (name) or integer (constant type).

Return value The method returns an object of the following type:

Type	Description
SimpleConstant ¹⁾	The activated type for the UserSignalValue object.

¹⁾ Refer to [SimpleConstant](#) on page 366.

Related topics

References

[Class Description \(UserSignalValue\)](#)..... 280

Classes for Configuring Conditions

Where to go from here

Information in this section

ConditionDistance	283
To get a distance condition.	
ConditionDistanceRefLine	283
To get/set a distance (along road reference line) condition.	
ConditionDrivenDistance	284
To get/set a driven distance condition.	
ConditionDuration	285
To get/set a duration condition.	
ConditionEndless	286
To get/set an endless condition.	
ConditionImmediate	287
To get/set an immediate condition.	
ConditionLateralAcceleration	287
To get a lateral acceleration condition (only for a maneuver).	
ConditionPosition	288
To get/set a position condition.	
ConditionTimegap	289
To get/set a time gap condition.	
ConditionTrigger	290
To get/set a trigger condition.	
ConditionVelocity	290
To get/set a velocity condition.	
Continue	291
To get information about the Continue lateral type.	
Continue	292
To get information about the Continue longitudinal type.	
TransitionsConditions	293
To get a transition conditions.	

Information in other sections

Overview of the Classes for Creating Scenarios	187
The overview gives you a quick overview of the classes for the Scenario Editor.	
Automating Scenarios in Python	180
You can change the parameters of the scenario.	

ConditionDistance

Purpose To get a distance condition.

Class Description (ConditionDistance)

Syntax `ConditionDistance = TransitionsConditions.Add(ConditionType)`

Purpose To get information about a distance condition.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.
Distance	Double	To get/set the distance.
Operator	ApproximationOperators ²⁾	To get the approximation operators.
Object1	RelativeVehicleSelection ³⁾	To get the reference vehicle of object 1.
Object2	RelativeVehicleSelection ³⁾	To get the reference vehicle of object 2.

¹⁾ Refer to [ConditionTypes](#) on page 385.

²⁾ Refer to [ApproximationOperators](#) on page 326.

³⁾ Refer to [RelativeVehicleSelection](#) on page 355.

Related topics

References

[TransitionsConditions.....](#) 293

ConditionDistanceRefLine

Purpose To get/set a distance (along road reference line) condition.

Class Description (ConditionDistanceRefLine)

Syntax

```
ConditionDistanceRefLine = TransitionsConditions.Add(ConditionType)
```

Purpose

To get/set a distance (along road reference line) condition.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.
Distance	Double	To get/set the distance.
Object1	RelativeVehicleSelection ²⁾	To get the first reference vehicle.
Object1ReferencePoint	ReferencePointTypes ³⁾	To get/set the reference point of object 1.
Object2	RelativeVehicleSelection ²⁾	To get the second reference vehicle.
Object2ReferencePoint	ReferencePointTypes ³⁾	To get/set the reference point of object 2.
Operator	ApproximationOperators ⁴⁾	To get the approximation operators.

¹⁾ Refer to [ConditionTypes](#) on page 385.

²⁾ Refer to [RelativeVehicleSelection](#) on page 355.

³⁾ Refer to [ReferencePointTypes](#) on page 388.

⁴⁾ Refer to [ApproximationOperators](#) on page 326.

Related topics

References

[TransitionsConditions.....](#) 293

ConditionDrivenDistance

Purpose

To get/set a driven distance condition.

Class Description (ConditionDrivenDistance)

Syntax

```
ConditionDistanceRefLine = TransitionsConditions.Add(ConditionType)
```

Purpose

To get information about a driven distance condition.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.
Distance	Double	To get/set the distance.
ReferenceVehicle	RelativeVehicleSelection ²⁾	To get the reference vehicle.

¹⁾ Refer to [ConditionTypes](#) on page 385.

²⁾ Refer to [RelativeVehicleSelection](#) on page 355.

Related topics

References

[TransitionsConditions.....](#) 293

ConditionDuration

Purpose

To get/set a duration condition.

Class Description (ConditionDuration)

Syntax

```
ConditionDuration = TransitionsConditions.Add(ConditionType)
```

Purpose

To get information about a duration condition.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.
Duration	Double	To get/set the duration.

¹⁾ Refer to [ConditionTypes](#) on page 385.

Related topics**References**

[TransitionsConditions.....](#) 293

ConditionEndless

Purpose

To get/set an endless condition.

Class Description (ConditionEndless)

Syntax

```
ConditionDuration = TransitionsConditions.Add(ConditionType)
```

Purpose

To get information about an endless condition.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.

¹⁾ Refer to [ConditionTypes](#) on page 385.

Related topics**References**

[TransitionsConditions.....](#) 293

ConditionImmediate

Purpose To get/set an immediate condition.

Class Description (ConditionImmediate)

Syntax `ConditionImmediate = TransitionsConditions.Add(ConditionType)`

Purpose To get information about an immediate condition.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.

¹⁾ Refer to [ConditionTypes](#) on page 385.

Related topics

References

[TransitionsConditions.....](#) 293

ConditionLateralAcceleration

Purpose To get a lateral acceleration condition (only for a maneuver).

Class Description (ConditionLateralAcceleration)

Syntax `ConditionLateralAcceleration = TransitionsConditions.Add(ConditionType)`

Purpose To get information about a lateral acceleration condition (only for a maneuver).

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Acceleration	Double	To get/set the acceleration.
Name	String	To get the condition name.
Operator	RelativeOperators ¹⁾	To get/set the relative operator.
Type	ConditionTypes ²⁾	To get the condition type.

¹⁾ Refer to [RelativeOperators](#) on page 353

²⁾ Refer to [ConditionTypes](#) on page 385.

Related topics

References

[TransitionsConditions..... 293](#)

ConditionPosition

Purpose To get/set a position condition.

Class Description (ConditionPosition)

Syntax `ConditionPosition = TransitionsConditions.Add(ConditionType)`

Purpose To get information about a position condition.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.

Attributes	Type	Purpose
Position	Double	To get/set the position.
ReferenceVehicle	RelativeVehicleSelection ²⁾	To get the reference vehicle.

¹⁾ Refer to [ConditionTypes](#) on page 385.

²⁾ Refer to [RelativeVehicleSelection](#) on page 355.

Related topics

References

[TransitionsConditions](#)..... 293

ConditionTimegap

Purpose To get/set a time gap condition.

Class Description (ConditionTimegap)

Syntax

```
ConditionTimegap = TransitionsConditions.Add(ConditionType)
```

Purpose To get information about a time gap condition.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.
Time	Double	To get/set the time gap.
Operator	ApproximationOperators ²⁾	To get the approximation operators.
Object1	RelativeVehicleSelection ³⁾	To get the reference vehicle of object 1.
Object2	RelativeVehicleSelection ³⁾	To get the reference vehicle of object 2.

¹⁾ Refer to [ConditionTypes](#) on page 385.

²⁾ Refer to [ApproximationOperators](#) on page 326.

³⁾ Refer to [RelativeVehicleSelection](#) on page 355.

Related topics**References**

[TransitionsConditions.....](#) 293

ConditionTrigger

Purpose

To get/set a trigger condition.

Class Description (ConditionTrigger)

Syntax

```
ConditionTrigger = TransitionsConditions.Add(ConditionType)
```

Purpose

To get information about a trigger condition.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.
Index	Integer	To get/set the index of the trigger.

¹⁾ Refer to [ConditionTypes](#) on page 385.

Related topics**References**

[TransitionsConditions.....](#) 293

ConditionVelocity

Purpose

To get/set a velocity condition.

Class Description (ConditionVelocity)

Syntax

```
ConditionTimegap = TransitionsConditions.Item(Index)
```

Purpose

To get information about a velocity condition.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the condition name.
Type	ConditionTypes ¹⁾	To get the condition type.
Velocity	Double	To get/set the velocity.
Operator	RelativeOperators ²⁾	To get the relative operators.
ReferenceVehicle	RelativeVehicleSelection ³⁾	To get the reference vehicle.

¹⁾ Refer to [ConditionTypes](#) on page 385.

²⁾ Refer to [RelativeOperators](#) on page 353.

³⁾ Refer to [RelativeVehicleSelection](#) on page 355.

Related topics

References

[TransitionsConditions.....](#) 293

Continue

Purpose

The class contains information about the **Continue** lateral type.

Class Description (Continue (LateralProfile))

Syntax

No direct creation

Purpose

To get information about the **Continue** lateral type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the lateral type.
Type	LateralProfileTypes ¹⁾	To get the lateral type constant.

¹⁾ Refer to [LateralProfileTypes](#) on page 387.

Continue

Purpose

To get information about the **Continue** longitudinal type.

Class Description (Continue (LongitudinalProfile))

Syntax

No direct creation

Purpose

To get information about the **Continue** longitudinal type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the longitudinal type.
Type	LongitudinalProfileTypes ¹⁾	To get the type constant of the longitudinal type.

¹⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

TransitionsConditions

Purpose To get transition conditions.

Where to go from here **Information in this section**

Class Description (TransitionsConditions)	293
To describe the class and its attributes.	
Add	294
To add a transition condition to the TransitionConditions collection.	
FindItem	295
To search for a condition type in the TransitionConditions collection.	
Item	296
To access a transition condition.	
Remove	297
To remove a transition condition.	

Class Description (TransitionsConditions)

Syntax

```
TransitionConditions = FellowTransition.Conditions
TransitionConditions = UserSignalTransition.Conditions
TransitionConditions = ManeuverTransition.Conditions
```

Purpose To get transition conditions.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of conditions.

Methods The class contains the following methods:

Method	Purpose
Add	To add a transition condition to the TransitionConditions collection. Refer to Add on page 294.
Item	To access a transition condition. Refer to Item on page 296.

Method	Purpose
FindItem	To search for a condition type in the TransitionConditions collection. Refer to FindItem on page 295.
Remove	To remove a transition condition. Refer to Remove on page 297.

Related topics

References

Class Description (FellowTransition)	255
Class Description (ManeuverTransition)	234
Class Description (UserSignalTransition)	279

Add

Class

TransitionsConditions

Syntax

```
TransitionCondition = TransitionsConditions.Add(ConditionType Type)
```

Purpose

To add a transition condition to the TransitionConditions collection.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Type	ConditionTypes ¹⁾	The type of the transition condition.

¹⁾ Refer to [ConditionTypes](#) on page 385.

Return value

The method returns an object of the following type:

Type	Description
ConditionLateralAcceleration ¹⁾ , ConditionDistance, ConditionDrivenDistance, ConditionDuration, ConditionEndless, ConditionImmediate, ConditionPosition, ConditionTimegap, ConditionTrigger, ConditionVelocity, or ConditionDistanceRefLine on page 283 ²⁾	The transition condition object.

¹⁾ Only for a ManeuverTransition

²⁾ Refer to [ConditionLateralAcceleration](#) on page 287, [ConditionDistance](#) on page 283, [ConditionDrivenDistance](#) on page 284, [ConditionDuration](#) on page 285, [ConditionEndless](#) on page 286, [ConditionImmediate](#) on page 287, [ConditionPosition](#)

Type	Description
on page 288, ConditionTimegap on page 289, ConditionTrigger on page 290, ConditionVelocity on page 290, or ConditionDistanceRefLine on page 283.	

Related topics**References**

[Class Description \(TransitionsConditions\)..... 293](#)

FindItem

Class

TransitionsConditions

Syntax

```
TransitionCondition = TransitionsConditions.FindItem(ConditionType Type,
int Index = 0)
```

Purpose

To search for a condition type in the TransitionConditions collection.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Type	ConditionTypes ¹⁾	The type of the transition condition.
Index	Integer	The index from which the search starts.

¹⁾ Refer to [ConditionTypes](#) on page 385.

Return value

The method returns an object of the following type:

Type	Description
ConditionDistance, ConditionDrivenDistance, ConditionDuration, ConditionEndless, ConditionImmediate, ConditionPosition, ConditionTimegap, ConditionTrigger, ConditionVelocity ¹⁾	The transition condition object.

¹⁾ Refer to [ConditionDistance](#) on page 283, [ConditionDrivenDistance](#) on page 284, [ConditionDuration](#) on page 285, [ConditionEndless](#) on page 286, [ConditionImmediate](#) on page 287, [ConditionPosition](#) on page 288, [ConditionTimegap](#) on page 289, [ConditionTrigger](#) on page 290, or [ConditionVelocity](#) on page 290.

Related topics

References

[Class Description \(TransitionsConditions\)..... 293](#)

Item

Class

TransitionsConditions

Syntax

```
TransitionCondition = TransitionsConditions.Item(Variant Index)
```

Purpose

To access a transition condition.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	Variant	The identifier (integer) of the transition condition.

Return value

The method returns an object of the following type:

Type	Description
ConditionLateralAcceleration ¹⁾ , ConditionDistance, ConditionDrivenDistance, ConditionDuration, ConditionEndless, ConditionImmediate, ConditionPosition, ConditionTimegap, ConditionTrigger, ConditionVelocity, or ConditionDistanceRefLine ²⁾	The transition condition object.

¹⁾ Only for a ManeuverTransition

²⁾ Refer to [ConditionLateralAcceleration](#) on page 287, [ConditionDistance](#) on page 283, [ConditionDrivenDistance](#) on page 284, [ConditionDuration](#) on page 285, [ConditionEndless](#) on page 286, [ConditionImmediate](#) on page 287, [ConditionPosition](#) on page 288, [ConditionTimegap](#) on page 289, [ConditionTrigger](#) on page 290, [ConditionVelocity](#) on page 290, or [ConditionDistanceRefLine](#) on page 283.

Related topics

References

[Class Description \(TransitionsConditions\)..... 293](#)

Remove

Class TransitionsConditions

Syntax `ReVal = TransitionsConditions.Remove(Variant Index)`

Purpose To remove a transition condition.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	Variant	The identifier of the transition condition to be removed. The identifier can be its index (integer type) or the TransitionCondition object.

Return value The method returns an object of the following type:

Type	Description
Boolean	True if successful

Related topics

References

[Class Description \(TransitionsConditions\)..... 293](#)

Classes for Model Configuration

Where to go from here

Information in this section

ListElements.....	298
To access a list of elements.	
ListElement.....	302
To access an element of a list.	
ManeuverConfiguration.....	303
To modify the maneuver configuration.	
ModelConfiguration.....	304
To modify the contents of the model configuration.	
ReadOnlyListElements.....	306
To access a read-only list of elements.	
ScenarioConfiguration.....	308
To modify the scenario configuration.	

Information in other sections

[Automation Using Python Scripts \(ModelDesk Automation !\[\]\(a870788d6ed9b8fd294b7654a8c8526b_img.jpg\)\)](#)
Gives information on how you can automate ModelDesk by using Python scripts.

ListElements

Purpose

To access a list of elements.

Where to go from here

Information in this section

Class Description (ListElements).....	299
To describe the class and its attributes.	
Add.....	300
To add an element.	
Item.....	300
To get a specific element of a list.	

Move.....	301
To move an element in the list.	
Remove.....	302
To remove an element from the list.	

Class Description (ListElements)

Syntax

```
FellowUserSignals = ScenarioConfiguration.FellowUserSignals
GlobalUserSignals = ScenarioConfiguration.GlobalUserSignals
Triggers = ScenarioConfiguration.Triggers
```

Purpose

To access a list of elements.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of elements in the list.

Methods

The class contains the following methods:

Method	Purpose
Add	To add an element. Refer to Add on page 300.
Item	To get a specific element. Refer to Item on page 300.
Move	To move an element in the list. Refer to Move on page 301.
Remove	To remove an element from the list. Refer to Remove on page 302.

Related topics

Basics

ScenarioConfiguration.....	308
--	---------------------

Add

Class ListElements

Syntax `RetVal = ListElements.Add()`

Purpose To add an element.

Parameters –

Return value The method returns the following parameter:

Type	Description
Boolean	True if successful

Related topics

References

[Class Description \(ListElements\)..... 299](#)

Item

Class ListElements

Syntax `ListElement = ListElements.Item(object Index)`

Purpose To get a specific element of a list.

Parameters The method uses the following parameters:

Parameter	Type	Description
Index	object	The element of the specific index.

Return value

The method returns the following parameter:

Type	Description
ListElement ¹⁾	The specific list element.

¹⁾ Refer to [ListElement](#) on page 302.

Related topics**References**

[Class Description \(ListElements\)..... 299](#)

Move

Class

ListElements

Syntax

```
ListElements.Move(object Value, integer NewIndex)
```

Purpose

To move an element in the list.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Value	Object	The element that is moved.
NewIndex	Integer	The position in the list to which the element is moved.

Return value

—

Related topics**References**

[Class Description \(ListElements\)..... 299](#)

Remove

Class ListElements

Syntax `RetVal = ListElements.Remove(ListElement Item)`

Purpose To remove an element from the list.

Parameters The method uses the following parameters:

Parameter	Type	Description
Item	ListElement ¹⁾	Element to be removed.

¹⁾ Refer to [ListElement](#) on page 302.

Return value The method returns the following parameter:

Type	Description
Boolean	True if successful.

Related topics

References

[Class Description \(ListElements\)..... 299](#)

ListElement

Purpose To access an element of a list.

Class Description (ListElement)

Syntax `ListElement = ListElements.Item(object Index)`
`ListElement = ReadOnlyListElements.Item(object Index)`

Purpose To get information on a list element.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Description	String	To get/set a description of the element.
Index	Integer	To get the index of the element.
Name	String	To get/set the name of the element.

Methods —

Related topics

References

ListElements.....	298
ReadOnlyListElements.....	306

ManeuverConfiguration

Purpose To modify the maneuver configuration.

Class Description (ManeuverConfiguration)

Syntax `ManeuverConfiguration = ModelConfiguration.ManeuverConfiguration`

Purpose To modify the maneuver configuration.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Triggers	ReadOnlyListElements ¹⁾	To get the collection of triggers.
UserSignals	ReadOnlyListElements ¹⁾	To get the collection of user signals.

¹⁾ Refer to [ReadOnlyListElements](#) on page 306.

Methods

—

Related topics**References**[ModelConfiguration..... 304](#)

ModelConfiguration

Purpose

To modify the contents of the model configuration.

Where to go from here**Information in this section**[Class Description \(ModelConfiguration\)..... 304](#)

To describe the class and its attributes.

[Reset..... 305](#)

To reset the model configuration.

Information in other sections[Environment Configuration Dialog..... 92](#)

To configure settings for the simulation model.

Class Description (ModelConfiguration)

Syntax`ModelConfiguration = Project.ModelConfiguration`

Purpose

To modify the contents of the model configuration.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ManeuverConfiguration	ManeuverConfiguration ¹⁾	To get the maneuver configuration.
ScenarioConfiguration	ScenarioConfiguration ²⁾	To get the scenario configuration.

¹⁾ Refer to [ManeuverConfiguration](#) on page 303.

²⁾ Refer to [ScenarioConfiguration](#) on page 308.

Methods

The class contains the following methods:

Method	Purpose
Reset	To reset the model configuration. Refer to Reset on page 305.

Related topics**References**

[Environment Configuration Dialog](#)..... 92
[Project \(ModelDesk Project and Experiment Management\)](#) 

Reset

Class

ModelConfiguration

Syntax

```
RetVal = ModelConfiguration.Reset()
```

Purpose

To reset the model configuration.

Parameters

—

Return value

The method returns the following parameter:

Type	Description
Boolean	True if successful.

Related topics**References**[Class Description \(ModelConfiguration\).....](#) 304

ReadOnlyListElements

Purpose

To access a read-only list of elements.

Where to go from here**Information in this section**[Class Description \(ReadOnlyListElements\).....](#) 306

To describe the class and its attributes.

[Item.....](#) 307

To get a specific element of the list.

[Move.....](#) 308

To move an element in the list.

Class Description (ReadOnlyListElements)

Syntax

```
ReadOnlyListElements = ManeuverConfiguration.Triggers  
ReadOnlyListElements = ManeuverConfiguration.UserSignals
```

Purpose

To access a read-only list of elements.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of elements in the list.

Methods

The class contains the following methods:

Method	Purpose
Item	To get a specific element. Refer to Item on page 307.
Move	To move an element in the list. Refer to Move on page 308.

Related topics**References**

[ManeuverConfiguration](#)..... 303

Item

Class

ReadOnlyListElements

Syntax

```
ListElement = ReadOnlyListElements.Item(object Index)
```

Purpose

To get a specific element of the list.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	object	Index of the specific element.

Return value

The method returns the following parameter:

Type	Description
ListElement ¹⁾	The specific list element.

¹⁾ Refer to [ListElement](#) on page 302.

Related topics**References**

[Class Description \(ReadOnlyListElements\)](#)..... 306

Move

Class ReadOnlyListElements

Syntax `ReadOnlyListElements.Move(object Value, integer NewIndex)`

Purpose To move an element in the list.

Parameters The method uses the following parameters:

Parameter	Type	Description
Value	object	The element to be moved.
NewIndex	Integer	The new position in the list to which the element is moved.

Return value —

Related topics

References

[Class Description \(ReadOnlyListElements\)..... 306](#)

ScenarioConfiguration

Purpose To modify the scenario configuration.

Class Description (ScenarioConfiguration)

Syntax `ScenarioConfiguration = ModelConfiguration.ScenarioConfiguration`

Purpose To modify the scenario configuration.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
FellowCount	Integer	To get the number of fellows.
FellowUserSignals	ListElements ¹⁾	To get the list of fellow user signals.
GlobalUserSignals	ListElements ¹⁾	To get the list of global user signals.
Triggers	ListElements ¹⁾	To get the list of triggers.

¹⁾ Refer to [ListElements](#) on page 298.

Methods

—

Related topics**References**

[ModelConfiguration](#)..... 304

Classes for Configuring Traffic Drivers

Where to go from here

Information in this section

DrivingStyle.....	310
To specify the driving style of a traffic driver.	
RuleCompliantDriving.....	311
To specify the rule compliant driving of a traffic driver.	
TrafficDrivers.....	313
To manage all the traffic driver of a ModelDesk project.	
TrafficDriverTypes.....	314
To activate a traffic driver in the maneuver or a fellow definition.	
TrafficDriver.....	316
To handle a traffic driver.	
TrafficSignRecognition.....	319
To specify the traffic sign recognition of a traffic driver.	

Information in other sections

Overview of the Object Model for Configuring Traffic Drivers.....	192
This gives you a quick overview of the classes for for configuring traffic drivers.	
Automating Scenarios in Python.....	180
You can change the parameters of the scenario.	

DrivingStyle

Purpose

To specify the driving style of a traffic driver.

Class Description (DrivingStyle)

Syntax

```
DrivingStyle = TrafficDriver.DrivingStyle
```

Purpose

To specify the driving style of a traffic driver.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
DistanceAtStandstill	Double	To get/set distance of the driver's vehicle to the vehicle ahead at standstill in meters
DistanceTime	Double	To get/set the distance of the driver's vehicle to the vehicle ahead in seconds while driving. This value is used to calculate the required distance in meters.
DetectionRange	Double	To get/set the range in front of the driver's vehicle in meters within other traffic participants are detected along the road reference line.
MaximumLongitudinalAcceleration	Double	To get/set the maximum acceleration in longitudinal direction in m/s^2 in everyday traffic.
ComfortableDeceleration	Double	To get/set the comfortable deceleration in m/s^2 in everyday traffic.
MaximumLateralAcceleration	Double	To get/set the maximum lateral acceleration in m/s^2 in curves.

Methods

—

Related topics**References**

Traffic Driver Dialog.....	101
TrafficDriver.....	316

RuleCompliantDriving

Purpose

To specify the rule compliant driving of a traffic driver.

Class Description (RuleCompliantDriving)

Syntax

```
RuleCompliantDriving = TrafficDriver.RuleCompliantDriving
```

Purpose

To specify the rule compliant driving of a traffic driver.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
DistanceAtStandstill	Double	To get/set the distance between the driver's vehicle and a traffic light or priority sign at standstill in meters.
SpeedLimitOffset	Double	To get/set the offset to the specified speed limit at which the driver's vehicle can drive slower or faster than the specified speed limit in km/h.
SignCompliantDrivingStart	Double	To get/set the offset from the position of the speed limit or no passing start sign to the position where the driver's vehicle complies with the traffic sign in meters.
SignCompliantDrivingEnd	Double	To get/set the offset from the position of the speed limit or no passing end sign to the position where the driver's vehicle complies with the traffic sign in meters.
YieldSpeedFactor	Double	To get/set the deceleration factor before junctions in percent. This value is used to calculate the velocity the vehicle slows down when approaching a junction where the right-of-way status must be checked.
SpeedLimitMode	SpeedLimitModes ¹⁾	To get/set the mode for road scenery speed limit use.

¹⁾ Refer to [SpeedLimitModes](#) on page 390.

Methods

—

Related topics**References**

Traffic Driver Dialog	101
TrafficDriver	316

TrafficDrivers

Purpose To manage all the traffic drivers of a ModelDesk project.

Where to go from here **Information in this section**

Class Description (TrafficDrivers).....	313
To describe the class and its attributes.	
Item.....	314
To get a specific traffic driver.	

Class Description (TrafficDrivers)

Syntax `TrafficDrivers = ActiveProject.TrafficDrivers`

Purpose To manage all the traffic drivers of the ModelDesk project.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of traffic drivers.
AvailableElements	String[]	To get the names of all available traffic drivers.

Methods The class contains the following methods:

Method	Purpose
Item	To get a specific traffic driver. Refer to Item on page 314.

Related topics

References

[Project \(ModelDesk Project and Experiment Management !\[\]\(b4eeff342f60cc7bcd67d869b4fedca2_img.jpg\)\)](#)

Item

Class TrafficDrivers

Syntax `TrafficDriver = TrafficDrivers.Item(object index)`

Purpose To get a specific traffic driver.

Parameters The method uses the following parameters:

Parameter	Type	Description
index	object	The index of the traffic driver.

Return value The method returns the following parameter:

Type	Description
TrafficDriver ¹⁾	The specific traffic driver.

¹⁾ Refer to [TrafficDriver](#) on page 316.

Related topics

References

[Class Description \(TrafficDrivers\)..... 313](#)

TrafficDriverTypes

Purpose To activate a traffic driver in the maneuver or a fellow definition.

Where to go from here

Information in this section

[Class Description \(TrafficDriverTypes\)..... 315](#)
To describe the class and its attributes.

[Activate..... 315](#)
To select the traffic driver to be activated.

Class Description (TrafficDriverTypes)

Syntax

```
TrafficDriverType = Maneuver.TrafficDriverType
```

or

```
TrafficDriverType = Fellow.TrafficDriverType
```

Purpose

To activate a traffic driver for the maneuver or a fellow definition.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
AvailableElements	String[]	To get a list of all the available traffic drivers.
ActiveElement	String	To get the name of the active traffic driver.

Methods

The class contains the following methods:

Method	Purpose
Activate	To select the traffic driver to be activated. Refer to Activate on page 315.

Related topics

References

Fellow	241
Maneuver	219

Activate

Class

TrafficDriverType

Syntax

```
RetVal = TrafficDriverType.Activate(object item)
```

Purpose

To select the traffic driver to be activated.

Parameters

The method uses the following parameters:

Parameter	Type	Description
item	object	The traffic driver.

Return value

The method returns the following parameter:

Type	Description
Boolean	True if successful.

Related topics**References**

[Class Description \(TrafficDriverTypes\)..... 315](#)

TrafficDriver

Purpose

To handle a traffic driver.

Where to go from here**Information in this section**

Class Description (TrafficDriver).....	316
To describe the class and its attributes.	
DiscardChanges.....	317
To discard changes.	
Save.....	318
To save the properties.	
SaveAs.....	319
To save the traffic driver under another file name.	

Class Description (TrafficDriver)

Syntax

```
TrafficDriver= TrafficDriver.Item()
```

Purpose To handle a traffic driver.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Author	String	To get/set the name of the author.
Description	String	To get/set the description.
DrivingStyle	DrivingStyle ¹⁾	To get/set the driving style.
Name	String	To get the name.
RuleCompliantDriving	RuleCompliantDriving ²⁾	To get/set the rule compliant driving.
TrafficSignRecognition	TrafficSignRecognition ³⁾	To get/set the traffic sign recognition.

¹⁾ Refer to [DrivingStyle](#) on page 310.

²⁾ Refer to [RuleCompliantDriving](#) on page 311.

³⁾ Refer to [TrafficSignRecognition](#) on page 319.

Methods The class contains the following methods:

Method	Purpose
DiscardChanges	To discard changes. Refer to DiscardChanges on page 317.
Save	To save the properties of the traffic driver. Refer to Save on page 318.
SaveAs	To save the traffic driver under another name. Refer to SaveAs on page 319.

Related topics

References

[TrafficDrivers..... 313](#)

DiscardChanges

Class TrafficDriver

Syntax `TrafficDriver.DiscardChanges()`

Purpose To discard changes.

Parameters

—

Return value

—

Related topics**References**[Class Description \(TrafficDriver\)..... 316](#)

Save

Class

TrafficDriver

Syntax`TrafficDriver.Save()`**Purpose**

To save the properties.

Parameters

—

Return value

The method returns the following parameter:

Type	Description
Boolean	True if successful.

Related topics**References**[Class Description \(TrafficDriver\)..... 316](#)

SaveAs

Class TrafficDriver

Syntax `TrafficDriver.SaveAs(FileName, OverwriteExisting)`

Purpose To save the traffic driver under another file name.

Parameters The method uses the following parameters:

Parameter	Type	Description
FileName	String	The file name.
OverwriteExisting	Boolean	Saving the traffic driver overwrites an existing file called FileName if true.

Return value The method returns the following parameter:

Type	Description
Boolean	True if successful.

Related topics

References

[Class Description \(TrafficDriver\)..... 316](#)

TrafficSignRecognition

Purpose To specify the traffic sign recognition of a traffic driver.

Class Description (TrafficSignRecognition)

Syntax `TrafficSignRecognition = TrafficDriver.TrafficSignRecognition`

Purpose

To specify the traffic sign recognition of a traffic driver.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
DistanceFront	Double	To get/set the distance for traffic sign recognition in front of the vehicle in meters.
DistanceRight	Double	To get/set the distance for traffic sign recognition to the right of the vehicle in meters.
DistanceLeft	Double	To get/set the distance for traffic sign recognition to the right of the vehicle in meters.
DistanceVertical	Double	To get/set the distance for traffic sign recognition above the vehicle in meters.
RelativeSignAngle	Double	To get/set the maximum angle between the traffic and the vector from the sign to the vehicle main point in degrees.

Methods

—

Related topics**References**

Traffic Driver Dialog.....	101
TrafficDriver.....	316

Miscellaneous Classes for Scenarios

Where to go from here

Information in this section

Acceleration.....	323
To get information about the Acceleration longitudinal type.	
AccelerationSource.....	324
To get information about the AccelerationSource type.	
ApproximationOperators.....	326
To get the approximation operator type.	
ApproximationOperator.....	327
To get information of an approximation operator.	
BasicRoad.....	328
To specify a BasicRoad profile.	
DependencyTypes.....	329
To select the dependency type.	
DependencyTypeAbsolute.....	330
To get information on an absolute dependency type.	
DependencyTypeRelative.....	331
To get information on a relative dependency type.	
DistanceMeter.....	332
To get information about the DistanceMeter longitudinal type.	
DistanceMeterSource.....	333
To get information about the DistanceMeterSource type.	
DistanceTime.....	334
To get information about the DistanceTime longitudinal type.	
DistanceTimeSource.....	335
To get information about the DistanceTimeSource type.	
FollowRoad.....	337
To specify a FollowRoad profile.	
LateralAcceleration.....	338
To specify a LateralAcceleration profile.	
LateralDeviation.....	339
To get information about the lateral deviation type of a fellow.	
LateralDeviationSource.....	339
To get the lateral deviation types of the fellow definition.	
LaneSelection.....	341
To get information about the lane selection of a fellow.	

LaneSelectionSource	342
To get the lane selection types of the fellow definition.	
NamedItem	344
To get information of a NamedItem object.	
NamedItemList	345
To access a NamedItemList object.	
NotUsed	348
To get information about the NotUsed lateral type.	
NotUsed	348
To get information about the NotUsed longitudinal type.	
PedalStimulus	349
To specify a PedalStimulus profile.	
Position	350
To get information about the Position longitudinal type.	
PositionSource	351
To get information about the PositionSource type.	
RelativeOperators	352
To get information on a relative operator.	
RelativeOperators	353
To select the relative operator type.	
RelativeObject	355
To get information of a relative object.	
RelativeVehicleSelection	355
To select the RelativeObject.	
Route	357
To get information of a route.	
RouteSelection	358
To select a route.	
SourceTypes	360
To specify the source type.	
Standstill	371
To specify a Standstill profile.	
StimulusAngle	371
To specify a StimulusAngle profile.	
StimulusTorque	372
To specify a StimulusTorque profile.	
Stop	373
To specify a Stop profile.	

StopWithin.....	374
To get information about the StopWithin longitudinal type.	
StopWithinSource.....	374
To get information about the StopWithinSource type.	
Table.....	376
To define a Table source type.	
TableValue.....	381
To access an entry of a Table source type.	
Velocity.....	382
To get information about the Velocity longitudinal type.	
VelocitySource.....	383
To get information about the VelocitySource type.	

Information in other sections

Overview of the Classes for Creating Scenarios.....	187
The overview gives you a quick overview of the classes for the Scenario Editor.	
Automating Scenarios in Python.....	180
You can change the parameters of the scenario.	

Acceleration

Purpose To get information about the **Acceleration** longitudinal type.

Class Description (Acceleration)

Syntax No direct creation

Purpose To get information about the **Acceleration** longitudinal type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
SourceType	AccelerationSource ¹⁾	To get the AccelerationSource object for defining acceleration parameters.
Name	String	To get the name of the longitudinal type.
Type	LongitudinalProfileTypes ²⁾	To get the type constant of the longitudinal type.

¹⁾ Refer to [AccelerationSource](#) on page 324.

²⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

AccelerationSource

Purpose

To get information about the **AccelerationSource** type.

Where to go from here**Information in this section**

Class Description (AccelerationSource)	324
To describe the class and its attributes.	
Activate	325
To activate a Acceleration source type.	

Class Description (AccelerationSource)

Syntax

No direct creation

Purpose

To get information about the **Acceleration** source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Ramp, Constant, or Table ¹⁾	To get the active Acceleration source type object.
AvailableElements	Strings (collection)	To get the names of all available Acceleration source types.

¹⁾ Refer to [Ramp](#) on page 366, [Constant](#) on page 362, or [Table](#) on page 370.

Methods

The class contains the following methods:

Method	Purpose
Activate	To activate an Acceleration source type. Refer to Activate on page 325.

Activate

Class

AccelerationSource

Syntax

```
AccelerationSourceType = AccelerationSource.Activate(Variant Type)
```

Purpose

To activate a Acceleration source type.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific source type. The value can be specified as string (name) or integer (constant type).

Return value

The method returns an object of the following type:

Type	Description
Ramp, Constant, or Table ¹⁾	A source type object according to the given parameter.

¹⁾ Refer to [Ramp](#) on page 366, [Constant](#) on page 362, or [Table](#) on page 370.

Related topics**References**

[Class Description \(AccelerationSource\)](#)..... 324

ApproximationOperators

Purpose To get the approximation operator type.

Where to go from here **Information in this section**

Class Description (ApproximationOperators).....	326
To describe the class and its attributes.	
Activate.....	327
To activate an approximation operator.	

Class Description (ApproximationOperators)

Syntax `ApproximationOperators = [Parent].ApproximationOperator`

Purpose To get the approximation operator type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	ApproximationOperator ¹⁾	To get the active approximation operator object.
AvailableElements	Strings (collection)	To get the names of all available approximation operators.

¹⁾ Refer to [ApproximationOperator](#) on page 327.

Methods The class contains the following methods:

Method	Purpose
Activate	To activate an approximation operator. Refer to Activate on page 327.

Activate

Class ApproximationOperators

Syntax `ApproximationOperator = ApproximationOperators.Activate(Variant Item)`

Purpose To activate an approximation operator.

Parameters The method uses the following parameters:

Parameter	Type	Description
Item	Variant	The identifier of the approximation operator. The value can be a string (name) or an integer (constant type).

Return value The method returns an object of the following type:

Type	Description
ApproximationOperator	The activated approximation operator object. Refer to ApproximationOperator on page 327

Related topics

References

[Class Description \(ApproximationOperators\)..... 326](#)

ApproximationOperator

Purpose To get information of an approximation operator.

Class Description (ApproximationOperator)

Syntax `ApproximationOperator = [Parent].ApproximationOperator`

Purpose To get information of an approximation operator.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of an approximation operator.
Type	ApproximationType ¹⁾	To get the approximation operator type.

¹⁾ Refer to [ApproximationType](#) on page 385.

BasicRoad

Purpose To specify a BasicRoad profile.

Class Description (BasicRoad)

Syntax No direct creation.

Purpose To specify a BasicRoad profile.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Straight or Circle ¹⁾	To get the active element.
AvailableElements	String[]	To get the names of all the available elements.
Name	String	To get the name of the BasicRoad profile.
Type	LateralProfileTypes ²⁾	To get the type of the lateral profile.

¹⁾ Refer to [Straight](#) on page 369 or [Circle](#) on page 361.

²⁾ Refer to [LateralProfileTypes](#) on page 387.

Methods

—

DependencyTypes

Purpose

To select the dependency type.

Where to go from here**Information in this section**

Class Description (DependencyTypes)	329
To describe the class and its attributes.	
Activate	330
To activate a dependency type.	

Class Description (DependencyTypes)

Syntax

DependencyType = [Parent].DependencyTypes

Purpose

To select the dependency type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	DependencyTypeAbsolute or DependencyTypeRelative ¹⁾	To get the active dependency type object.
AvailableElements	Strings (collection)	To get the names of all available dependency types.

¹⁾ Refer to [DependencyTypeAbsolute](#) on page 330 or [DependencyTypeRelative](#) on page 331.

Methods

The class contains the following methods:

Method	Purpose
Activate	To activate a dependency type. Refer to Activate on page 356.

Activate

Class DependencyTypes

Syntax `DependencyType = DependencyTypes.Activate(Variant Item)`

Purpose To activate a dependency type.

Parameters The method uses the following parameters:

Parameter	Type	Description
Item	Variant	The identifier of the dependency type. You can use strings ('absolute', 'relative') or type constant (see DependencyTypeConstants on page 386).

Return value The method returns an object of the following type:

Type	Description
DependencyTypeAbsolute or DependencyTypeRelative ¹⁾	The activated dependency type object.

¹⁾ Refer to [DependencyTypeAbsolute](#) on page 330 or [DependencyTypeRelative](#) on page 331.

Related topics

References

[Class Description \(DependencyTypes\)](#)..... 329

DependencyTypeAbsolute

Purpose To get information on an absolute dependency type.

Class Description (DependencyTypeAbsolute)

Syntax No direct creation

Purpose To get information on an absolute dependency type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the dependency type.
Type	DependencyTypeConstants ¹⁾	To get the type of the dependency type.

¹⁾ Refer to [DependencyTypeConstants](#) on page 386.

DependencyTypeRelative

Purpose To get information on a relative dependency type.

Class Description (DependencyTypeRelative)

Syntax No direct creation

Purpose To get information on a relative dependency type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
RelativeVehicle	RelativeVehicleSelection ¹⁾	To get the relative vehicle selection object.
Name	String	To get the name of the dependency type.
Type	DependencyTypeConstants ²⁾	To get the type of the dependency type.

¹⁾ Refer to [RelativeVehicleSelection](#) on page 355.

²⁾ Refer to [DependencyTypeConstants](#) on page 386.

DistanceMeter

Purpose To get information about the **DistanceMeter** longitudinal type.

Class Description (DistanceMeter)

Syntax No direct creation

Purpose To get information about the **DistanceMeter** longitudinal type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the longitudinal type.
PointOnReferenceVehicle	ReferencePointTypes ¹⁾	To get/set the reference point on the referenced vehicle. Valid values are front, rear, and main.
ReferencePoint	ReferencePointTypes ¹⁾	To get/set the reference point on the ASM vehicle. Valid values are front, rear, and main.
RelativeVehicle	RelativeVehicleSelection ²⁾	To get the RelativeVehicleSelection object.
SourceType	DistanceMeterSource ³⁾	To get the DistanceMeterSource object for defining distance parameters.
Type	LongitudinalProfileTypes ⁴⁾	To get the type constant of the longitudinal type.

¹⁾ Refer to [ReferencePointTypes](#) on page 388.

²⁾ Refer to [RelativeVehicleSelection](#) on page 355.

³⁾ Refer to [DistanceMeterSource](#) on page 333.

⁴⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

DistanceMeterSource

Purpose To get information about the `DistanceMeterSource` type.

Where to go from here Information in this section

Class Description (DistanceMeterSource).....	333
To describe the class and its attributes.	
Activate.....	334
To activate a DistanceMeter source type.	

Class Description (DistanceMeterSource)

Syntax No direct creation

Purpose To get information about the `DistanceMeter` source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Constant, Ramp, Extern, or FinalDistance ¹⁾	To get the active DistanceMeter source type object.
AvailableElements	Strings (collection)	To get the names of all available DistanceMeter source types.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Extern](#) on page 362, or [FinalDistance](#) on page 363.

Methods The class contains the following methods:

Method	Purpose
Activate	To activate a DistanceMeter source type. Refer to Activate on page 334.

Activate

Class DistanceTimeSource

Syntax `DistanceMeterSourceType = DistanceMeterSource.Activate(Variant Type)`

Purpose To activate a DistanceMeter source type.

Parameters The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific source type. The value can be specified as string (name) or integer (constant type).

Return value The method returns an object of the following type:

Type	Description
Constant, Ramp, Extern, or FinalDistance ¹⁾	A source type object according to the given parameter.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Extern](#) on page 362, or [FinalDistance](#) on page 363.

Related topics

References

[Class Description \(DistanceTimeSource\)..... 336](#)

DistanceTime

Purpose To get information about the **DistanceTime** longitudinal type.

Class Description (DistanceTime)

Syntax No direct creation

Purpose To get information about the `DistanceTime` longitudinal type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
SourceType	DistanceTimeSource ¹⁾	To get the DistanceTimeSource object for defining distance parameters.
RelativeVehicle	RelativeVehicleSelection ²⁾	To get the RelativeVehicleSelection object.
Name	String	To get the name of the longitudinal type.
Type	LongitudinalProfileTypes ³⁾	To get the type constant of the longitudinal type.

¹⁾ Refer to [DistanceTimeSource](#) on page 335.

²⁾ Refer to [RelativeVehicleSelection](#) on page 355.

³⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

DistanceTimeSource

Purpose To get information about the `DistanceTimeSource` type.

Where to go from here Information in this section

Class Description (DistanceTimeSource)	336
To describe the class and its attributes.	
Activate	336
To activate a DistanceTime source type.	

Class Description (DistanceTimeSource)

Syntax No direct creation

Purpose To get information about the **DistanceTime** source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Constant, Ramp, Extern, FinalDistance, or Table ¹⁾	To get the active DistanceTime source type object.
AvailableElements	Strings (collection)	To get the names of all available DistanceTime source types.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Extern](#) on page 362, [FinalDistance](#) on page 363, or [Table](#) on page 370.

Methods The class contains the following methods:

Method	Purpose
Activate	To activate a DistanceTime source type. Refer to Activate on page 336.

Activate

Class DistanceTimeSource

Syntax `DistanceTimeSourceType = DistanceTimeSource.Activate(Variant Type)`

Purpose To activate a DistanceTime source type.

Parameters The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific source type. The value can be specified as string (name) or integer (constant type).

Return value

The method returns an object of the following type:

Type	Description
Constant, Ramp, Extern, FinalDistance, or Table ¹⁾	A source type object according to the given parameter.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Extern](#) on page 362, [FinalDistance](#) on page 363, or [Table](#) on page 370.

Related topics**References**

[Class Description \(DistanceTimeSource\)..... 336](#)

FollowRoad

Purpose

To specify a FollowRoad profile.

Class Description (FollowRoad)

Syntax

No direct creation.

Purpose

To specify a FollowRoad profile.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
DriverTolerance	Double	To get/set the value of the driver tolerance.
DriverToleranceEnabled	Boolean	To get/set the use of the driver tolerance.
DriverType	DriverTypes ¹⁾	To get/set the driver type.
DriverTypeValues	DriverType ²⁾	To get the values of the driver type.
ForceToRoad	Boolean	To get/set the ForceToRoad flag.
LaneIndex	Constant, FinalLane, Table, Manual, or Extern ³⁾	To get the lane index.

Attributes	Type	Purpose
LateralOffset	Constant, Table, Manual, or Extern ⁴⁾	To get the lateral offset.
Name	String	To get the name of the profile.
ReferenceType	ReferencedVehicleTypes ⁵⁾	To get/set the reference type (relative to lane = 0 or relative to object = 1)
Type	LateralProfileTypes ⁶⁾	To get the type of the lateral profile.

¹⁾ Refer to [DriverTypes](#) on page 386.

²⁾ Refer to [DriverType](#) on page 213.

³⁾ Refer to [Constant](#) on page 362, [FinalLane](#) on page 363, [Table](#) on page 370, [Manual](#) on page 365, or [Extern](#) on page 362.

⁴⁾ Refer to [Constant](#) on page 362, [Table](#) on page 370, [Manual](#) on page 365, or [Extern](#) on page 362.

⁵⁾ Refer to [ReferencedVehicleTypes](#) on page 388.

⁶⁾ Refer to [LateralProfileTypes](#) on page 387.

Methods

—

LateralAcceleration

Purpose

To specify a LateralAcceleration profile.

Class Description (LateralAcceleration)

Syntax

No direct creation

Purpose

To specify a LateralAcceleration profile.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the LateralAcceleration profile.
SourceType	Constant, Table, or Ramp ¹⁾	To get the source type for the acceleration pedal.

Attributes	Type	Purpose
Type	LongitudinalProfileTypes ²⁾	To get the longitudinal profile type.

¹⁾ Refer to [Constant](#) on page 362, [Table](#) on page 370, or [Ramp](#) on page 366.

²⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

Methods

—

LateralDeviation

Purpose

To get information about the lateral deviation type of a fellow.

Class Description (LateralDeviation)

Syntax

No direct creation

Purpose

To get information about the lateral deviation type of a fellow.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the lateral type.
Type	LateralProfileTypes ¹⁾	To get the lateral type constant.
DependencyType	DependencyType ²⁾	To get the DependencyType object.
SourceType	LateralDeviationSource ³⁾	To get the LateralDeviationSource object.

¹⁾ Refer to [LateralProfileTypes](#) on page 387.

²⁾ Refer to [DependencyTypes](#) on page 329.

³⁾ Refer to [LateralDeviationSource](#) on page 339.

LateralDeviationSource

Purpose

The class contains the lateral deviation types of the fellow definition.

Where to go from here**Information in this section**

Class Description (LateralDeviationSource).....	340
To describe the class and its attributes.	
Activate.....	341
To get a specific lateral deviation type.	

Class Description (LateralDeviationSource)

Syntax

No direct creation

Purpose

To get information about the lateral deviation type of a fellow.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Constant, Ramp, Extern, FinalDeviation, SmoothLaneChange, or Table ¹⁾	To get the deviation from the center line.
AvailableElements	Strings (collection)	To get the names of all available lateral deviation types.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Extern](#) on page 362, [FinalDeviation](#) on page 362, [SmoothLaneChange](#) on page 368, or [Table](#) on page 370.

Methods

The class contains the following methods:

Method	Purpose
Activate	To get a specific lateral deviation type. Refer to Activate on page 341.

Activate

Class LateralDeviationSource

Syntax `LateralDeviationType = LateralDeviationSource.Activate(Variant Type)`

Purpose To get a specific lateral deviation source.

Parameters The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific lateral deviation source. The value can be specified as string (name) or integer (constant type, see SourceTypeConstants on page 389).

Return value The method returns an object of the following type:

Type	Description
Constant, Ramp, Extern, FinalDeviation, SmoothLaneChange, or Table ¹⁾	A lateral deviation source object according to the given parameter.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Extern](#) on page 362, [FinalDeviation](#) on page 362, or [SmoothLaneChange](#) on page 368, or [Table](#) on page 370.

Related topics

References

[Class Description \(LateralDeviationSource\)..... 340](#)

LaneSelection

Purpose To get information about the lane selection of a fellow.

Class Description (LaneSelection)

Syntax No direct creation

Purpose To get information about the lane selection type of a fellow.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the lateral type.
Type	LateralProfileTypes ¹⁾	To get the lateral type constant.
DependencyType	DependencyType ²⁾	To get the DependencyType object.
SourceType	LaneSelectionSource ³⁾	To get the LaneSelectionSource object.
ForceToRoad	Bool	To get/set the ForceToRoad state.

¹⁾ Refer to [LateralProfileTypes](#) on page 387.

²⁾ Refer to [DependencyTypes](#) on page 329.

³⁾ Refer to [LaneSelectionSource](#) on page 342.

LaneSelectionSource

Purpose The class contains the lane selection types of the fellow definition.

Where to go from here

Information in this section

[Class Description \(LaneSelectionSource\)](#)..... 343

To describe the class and its attributes.

[Activate](#)..... 343

To get a specific lane selection type.

Class Description (LaneSelectionSource)

Syntax No direct creation

Purpose To get information about the lane selection type of a fellow.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Constant, LaneChange, SmoothLaneChange, Extern, FinalLane, or Table ¹⁾	To get the lane selection type.
AvailableElements	Strings (collection)	To get the names of all available lane selection types.

¹⁾ Refer to [Constant](#) on page 362, [LaneChange](#) on page 365, [SmoothLaneChange](#) on page 368, [Extern](#) on page 362, [FinalLane](#) on page 363, or [Table](#) on page 370.

Methods The class contains the following methods:

Method	Purpose
Activate	To get a specific lane selection type. Refer to Activate on page 343.

Activate

Class LaneSelectionSource

Syntax `LaneSelectionType = LaneSelectionSource.Activate(Variant Type)`

Purpose To get a specific lane selection source.

Parameters The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific lane selection source. The value can be specified as string (name) or integer (constant type, see SourceTypeConstants on page 389).

Return value

The method returns an object of the following type:

Type	Description
Constant, LaneChange, SmoothLaneChange, Extern, FinalLane, or Table ¹⁾	A lane selection deviation source object according to the given parameter.

¹⁾ Refer to [Constant](#) on page 362, [LaneChange](#) on page 365, [SmoothLaneChange](#) on page 368, [Extern](#) on page 362, [FinalLane](#) on page 363, or [Table](#) on page 370.

Related topics**References**

[Class Description \(LaneSelectionSource\)](#)..... 343

NamedItem

Purpose

To get information of a NamedItem object.

Class Description (NamedItem)

Syntax

```
NamedItem = NamedItemList.Item()
```

Purpose

To access a NamedItem object.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name.
Description	String	To get the description.

Related topics**References**

[Class Description \(NamedItemList\)](#)..... 345

NamedItemList

Purpose To access a NamedItemList object.

Where to go from here Information in this section

Class Description (NamedItemList)	345
To describe the class and its attributes.	
Add	346
To add a NamedItem object.	
Item	346
To access a specific NamedItem object.	
Remove	347
To remove a NamedItem object.	

Class Description (NamedItemList)

Syntax No direct creation

Purpose To access a NamedItemList object.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	The number of NamedItem objects.

Methods The class contains the following methods:

Method	Purpose
Add	To add a NamedItem object. Refer to Add on page 346.
Item	To access a specific NamedItem object. Refer to Item on page 346.
Remove	To remove a NamedItem object. Refer to Remove on page 347.

Add

Class NamedItemList

Syntax `NamedItem = NamedItemList.Add(String Name)`

Purpose To add a NamedItem object.

Parameters The method uses the following parameters:

Parameter	Type	Description
Name	String	Name of the added NamedItem object.

Return value The method returns an object of the following type:

Type	Description
NamedItem ¹⁾	The added NamedItem object.

¹⁾ Refer to [NamedItem](#) on page 344.

Related topics

References

[Class Description \(NamedItemList\)..... 345](#)

Item

Class NamedItemList

Syntax `NamedItem = NamedItemList.Item(object Index)`

Purpose To access a specific NamedItem object.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	object	The index of the specific NamedItem object.

Return value

The method returns an object of the following type:

Type	Description
NamedItem ¹⁾	The specific NamedItem object.

¹⁾ Refer to [NamedItem](#) on page 344.

Related topics**References**

[Class Description \(NamedItemList\).....](#) 345

Remove

Class

NamedItemList

Syntax

```
RetVal = NamedItemList.Item(object Index)
```

Purpose

To remove a NamedItem object.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Index	object	The index of the NamedItem object to be removed.

Return value

The method returns an object of the following type:

Type	Description
Boolean	True if successful.

Related topics**References**

[Class Description \(NamedItemList\)..... 345](#)

NotUsed

Purpose

The class contains information about the **NotUsed** lateral type.

Class Description (NotUsed (LateralProfile))

Syntax

No direct creation

Purpose

To get information about the **NotUsed** lateral type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the lateral type.
Type	LateralProfileTypes ¹⁾	To get the lateral type constant.

¹⁾ Refer to [LateralProfileTypes](#) on page 387.

NotUsed

Purpose

The class contains information about the **NotUsed** longitudinal type.

Class Description (NotUsed (LongitudinalProfile))

Syntax No direct creation

Purpose To get information about the **NotUsed** longitudinal type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the longitudinal type.
Type	LongitudinalProfileTypes ¹⁾	To get the longitudinal type constant.

¹⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

PedalStimulus

Purpose To specify a PedalStimulus profile.

Class Description (PedalStimulus)

Syntax No direct creation

Purpose To specify a PedalStimulus profile.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
AccelerationPedal	Manual, Extern, Constant, Table, or Ramp ¹⁾	To get the source type for the acceleration pedal.
BrakePedal	Manual, Extern, Constant, Table, or Ramp ¹⁾	To get the source type for the brake pedal.
Name	String	To get the name of the PedalStimulus profile.

Attributes	Type	Purpose
Type	LongitudinalProfileTypes ²⁾	To get the longitudinal profile type.

¹⁾ Refer to [Manual](#) on page 365, [Extern](#) on page 362, [Constant](#) on page 362, [Table](#) on page 370, or [Ramp](#) on page 366.

²⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

Methods

—

Position

Purpose

To get information about the **Position** longitudinal type.

Class Description (Position)

Syntax

No direct creation

Purpose

To get information about the **Position** longitudinal type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
SourceType	PositionSource ¹⁾	To get the PositionSource object for defining distance parameters.
Name	String	To get the name of the longitudinal type.
Type	LongitudinalProfileTypes ²⁾	To get the type constant of the longitudinal type.

¹⁾ Refer to [PositionSource](#) on page 351.

²⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

PositionSource

Purpose To get information about the **PositionSource** type.

Where to go from here Information in this section

Class Description (PositionSource).....	351
To describe the class and its attributes.	
Activate.....	352
To activate a position source type.	

Class Description (PositionSource)

Syntax No direct creation

Purpose To get information about the **Position** source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	YawAngleConstant, Ramp, Extern, or Table ¹⁾	To get the active position source type object.
AvailableElements	Strings (collection)	To get the names of all available position source types.

¹⁾ Refer to [YawAngleConstant](#) on page 370, [Ramp](#) on page 366, [Extern](#) on page 362, or [Table](#) on page 370.

Methods The class contains the following methods:

Method	Purpose
Activate	To activate a position source type. Refer to Activate on page 352.

Activate

Class PositionSource

Syntax `PositionSourceType = PositionSource.Activate(Variant Type)`

Purpose To activate a position source type.

Parameters The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific source type. The value can be specified as string (name) or integer (constant type).

Return value The method returns an object of the following type:

Type	Description
YawAngleConstant, Ramp, Extern, or Table ¹⁾	A source type object according to the given parameter.

¹⁾ Refer to [YawAngleConstant](#) on page 370, [Ramp](#) on page 366, [Extern](#) on page 362, or [Table](#) on page 370.

Related topics

References

[Class Description \(PositionSource\)](#)..... 351

RelativeOperators

Purpose To get information on a relative operator.

Class Description (RelativeOperator)

Syntax `RelativeOperator = [Parent].RelativeOperator`

Purpose To get information on a relative operator.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the relative operator ('<=' or '>=').
Type	RelativeOperatorType ¹⁾	To get the type constant of relative operator.

¹⁾ Refer to [RelativeOperatorType](#) on page 389.

RelativeOperators

Purpose To select the relative operator type.

Where to go from here

Information in this section

Class Description (RelativeOperators).....	353
To describe the class and its attributes.	
Activate.....	354
To activate a relative operator.	

Class Description (RelativeOperators)

Syntax

```
RelativeOperators = [Parent].RelativeOperators
```

Purpose To select the relative operator type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	RelativeOperators ¹⁾	To get the active relative operator object.

Attributes	Type	Purpose
AvailableElements	Strings (collection)	To get the names of all available relative operators.

¹⁾ Refer to [RelativeOperators](#) on page 352.

Methods

The class contains the following methods:

Method	Purpose
Activate	To activate a relative operator. Refer to Activate on page 354.

Activate

Class RelativeOperators

Syntax `RelativeOperator = RelativeOperators.Activate(Variant Item)`

Purpose To activate a relative operator.

Parameters The method uses the following parameters:

Parameter	Type	Description
Item	Variant	The identifier of the relative operator. Valid values are integer (constant type) and string(name).

Return value The method returns an object of the following type:

Type	Description
RelativeOperators ¹⁾	The activated relative operator object.

¹⁾ Refer to [RelativeOperators](#) on page 352.

Related topics

References

[Class Description \(RelativeOperators\)..... 353](#)

RelativeObject

Purpose To get information of a relative object.

Class Description (RelativeObject)

Syntax `RelativeObject = [Parent].RelativeObject`

Purpose To get information on a relative object.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of a relative object.
Index	Integer	To get the index of a relative object.
Type	RelativeObjectType ¹⁾	To get the relative object type.
Valid	Boolean	True if valid

¹⁾ Refer to [RelativeObjectType](#) on page 389.

RelativeVehicleSelection

Purpose To select the RelativeObject.

Where to go from here Information in this section

Class Description (RelativeVehicleSelection)	356
To describe the class and its attributes.	
Activate	356
To activate a relative vehicle.	

Class Description (RelativeVehicleSelection)

Syntax

```
RelativeVehicleSelection = [Parent].RelativeVehicleSelection
```

Purpose

To select the RelativeObject.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	RelativeObject ¹⁾	To get the active relative object (vehicle or position marker).
AvailableElements	Strings (collection)	To get the names of all available relative objects.

¹⁾ Refer to [RelativeObject](#) on page 355.

Methods

The class contains the following methods:

Method	Purpose
Activate	To activate a relative object. Refer to Activate on page 356.

Activate

Class

RelativeVehicleSelection

Syntax

```
RelativeVehicle = RelativeVehicleSelection.Activate(Variant Item)
```

Purpose

To activate a relative object.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Item	Variant	The identifier of the relative vehicle. Only strings are valid. The initial substring of the complete fellow names are allowed, for example, 'M1', 'F1', or 'F3'.

Return value

The method returns an object of the following type:

Type	Description
RelativeObject ¹⁾	The activated relative vehicle object.

¹⁾ Refer to [RelativeObject](#) on page 355.

Related topics**References**

[Class Description \(RelativeVehicleSelection\)..... 356](#)

Route

Purpose

To get information of a route.

Class Description (Route)

Syntax

`Route = [Parent].Route`

Purpose

To get information on a route.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the route.
Description	String	To get the description of the route.
Valid	Boolean	To get the valid state of the route.
ReversePossible	Boolean	To get the information whether the route can be used in reverse direction.
Index	Long	To get the index of the route.

RouteSelection

Purpose To select a route.

Where to go from here Information in this section

Class Description (RouteSelection).....	358
To describe the class and its attributes.	
Activate.....	359
To activate a route.	

Class Description (RouteSelection)

Syntax

```
RouteSelection = ManeuverSequence.RouteSelection
RouteSelection = FellowSequence.RouteSelection
```

Purpose To select the route.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Route ¹⁾	To get the active route object.
AvailableElements	Strings (collection)	To get the names of all available routes.
Direction	DrivingDirectionConstants ²⁾	To get/set the direction of travel.
UseExternal	Boolean	To get/set the flag to indicate that external route definitions are used.

¹⁾ Refer to [Route](#) on page 357.

²⁾ Refer to [DrivingDirectionConstants](#) on page 387.

Methods The class contains the following methods:

Method	Purpose
Activate	To activate a route. Refer to Activate on page 359.

Related topics**References**

FellowSequence.....	251
ManeuverSequence.....	229

Activate

Class

RouteSelection

Syntax`Route = RouteSelection.Activate(Variant Item)`**Purpose**

To activate a route.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Item	Variant	The identifier of the route. Only strings are valid. The initial substring of the complete route names are allowed, for example, 'R1', or 'R2'.

Return value

The method returns an object of the following type:

Type	Description
Route ¹⁾	The activated route object.

¹⁾ Refer to [Route](#) on page 357.**Related topics****References**

Class Description (RouteSelection).....	358
---	---------------------

SourceTypes

Purpose To specify the source type.

Where to go from here

Information in this section

Circle.....	361
To define a Circle source type.	
Constant.....	362
To define a Constant source type.	
Extern.....	362
To define an Extern source type.	
FinalDeviation.....	362
To define a FinalDeviation source type.	
FinalDistance.....	363
To define a FinalDistance source type.	
FinalLane.....	363
To define a FinalLane source type.	
FinalVelocity.....	364
To define a FinalVelocity source type.	
Fixed.....	364
To define a Fixed source type.	
LaneChange.....	365
To define a LaneChange source type.	
Manual.....	365
To define a Manual source type.	
Pulse.....	365
To define a Pulse source type.	
Ramp.....	366
To define a Ramp source type.	
SimpleConstant.....	366
To define a SimpleConstant source type.	
Sine.....	367
To define a Sine source type.	
SineWithDwell.....	367
To define a SineWithDwell source type.	
SmoothLaneChange.....	368
To define a SmoothLaneChange source type.	

Step.....	368
To define a Step source type.	
StopWithin.....	369
To define a StopWithin source type.	
Straight.....	369
To define a Straight source type.	
Table.....	370
To define a Table source type.	
UntilStop.....	370
To define an UntilStop source type.	
YawAngleConstant.....	370
To define a YawAngleConstant source type.	

Information in other sections

Source Properties.....	172
The source properties are used in several properties to specify the kind of source.	

Circle

Syntax	<code>Circle = [Parent].Circle</code>
---------------	---------------------------------------

Purpose	To define a Circle source type.
----------------	---------------------------------

Attributes	The class contains the following attributes:
-------------------	--

Attributes	Type	Purpose
Radius	Double	To get/set the radius in meters.
Direction	DirectionTypes	To get/set the direction.
ControlMode	ControlModeTypes ¹⁾	To get/set the control mode.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

¹⁾ Refer to [ControlModeTypes](#) on page 386.

Constant

Syntax `Constant = [Parent].Constant`

Purpose To define a Constant source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Constant	Double	To get/set the constant value.
HoldCurrentValue	Boolean	To get/set the hold current value flag.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

Extern

Syntax `Extern = [Parent].Extern`

Purpose To define an Extern source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

FinalDeviation

Syntax `FinalDeviation = [Parent].FinalDeviation`

Purpose To define a FinalDeviation source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Deviation	Double	To get/set the deviation value.
LateralVelocity	Double	To get/set the lateral velocity value.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

FinalDistance

Syntax

`FinalDistance = [Parent].FinalDistance`

Purpose

To define a FinalDistance source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Distance	Double	To get/set the distance value.
Duration	Double	To get/set the duration value.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

FinalLane

Syntax

`FinalLane = [Parent].FinalLane`

Purpose

To define a FinalLane source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
FinalLane	Double	To get/set the final lane value.
LateralVelocity	Double	To get/set the lateral velocity value.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

FinalVelocity

Syntax `FinalVelocity = [Parent].FinalVelocity`

Purpose To define a FinalVelocity source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Velocity	Double	To get/set the velocity value.
NegativeAcceleration	Double	To get/set the negative acceleration value.
PositiveAcceleration	Double	To get/set the positive acceleration value.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

Fixed

Syntax `Fixed = [Parent].Fixed`

Purpose To define a Fixed source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

LaneChange

Syntax LaneChange = [Parent].LaneChange

Purpose To define a LaneChange source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
StartLane	Double	To get/set the start lane value.
EndLane	Double	To get/set the end lane value.
Duration	Double	To get/set the duration value.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

Manual

Syntax Manual = [Parent].Manual

Purpose To define a Manual source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

Pulse

Syntax Pulse = [Parent].Pulse

Purpose To define a Pulse source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
SteeringWheelAngle	Double	To get/set the angle of the steering wheel.
Direction	DirectionTypes ¹⁾	To get/set the direction.
InterpretationType	InterpretationTypes ²⁾	To get/set the interpretation type.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.
X1	Double	To get/set the X 1 value.
X2	Double	To get/set the X 2 value.
X3	Double	To get/set the X 3 value.
X4	Double	To get/set the X 4 value.

¹⁾ Refer to [DirectionTypes](#) on page 386.

²⁾ Refer to [InterpretationTypes](#) on page 387.

Ramp

Syntax

`Ramp = [Parent].Ramp`

Purpose

To define a Ramp source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Start	Double	To get/set the start value.
End	Double	To get/set the end value.
Duration	Double	To get/set the duration value.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

SimpleConstant

Syntax

`SimpleConstant = [Parent].SimpleConstant`

Purpose

To define a SimpleConstant source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Constant	Double	To get/set the constant value.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

Sine

Syntax

`Sine = [Parent].Sine`

Purpose

To define a Sine source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Amplitude	Double	To get/set the amplitude value.
StartFrequency	Double	To get/set the start frequency value.
EndFrequency	Double	To get/set the end frequency value.
Duration	Double	To get/set the duration value.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

SineWithDwell

Syntax

`SineWithDwell = [Parent].SineWithDwell`

Purpose

To define a SineWithDwell source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
SteeringWheelAngle	Double	To get/set the angle of the steering wheel.
Direction	DirectionTypes ¹⁾	To get/set the direction.
InterpretationType	InterpretationTypes ²⁾	To get/set the interpretation type.

Attributes	Type	Purpose
Name	String	To get the source type name.
Type	Integer	To get the source type constant.
X1	Double	To get/set the X 1 value.
X2	Double	To get/set the X 2 value.

¹⁾ Refer to [DirectionTypes](#) on page 386.

²⁾ Refer to [InterpretationTypes](#) on page 387.

SmoothLaneChange

Syntax `SmoothLaneChange = [Parent].SmoothLaneChange`

Purpose To define a SmoothLaneChange source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
StartDeviation	Double	To get/set the start deviation value.
EndDeviation	Double	To get/set the end deviation value.
Duration	Double	To get/set the duration value.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

Step

Syntax `Step = [Parent].Step`

Purpose To define a Step source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
SteeringWheelAngle	Double	To get/set the angle of the steering wheel.
Direction	DirectionTypes ¹⁾	To get/set the direction.
InterpretationType	InterpretationTypes ²⁾	To get/set the interpretation type.

Attributes	Type	Purpose
Name	String	To get the source type name.
Type	Integer	To get the source type constant.
X1	Double	To get/set the X 1 value.
X2	Double	To get/set the X 2 value.

¹⁾ Refer to [DirectionTypes](#) on page 386.

²⁾ Refer to [InterpretationTypes](#) on page 387.

StopWithin

Syntax `StopWithin = [Parent].StopWithin`

Purpose To define a StopWithin source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
HoldCurrentValue	Boolean	To get/set the hold current value flag.
ApproachVelocity	Double	To get/set the approach velocity value.
StopDistance	Double	To get/set the stop distance value.
Deceleration	Double	To get/set the deceleration value.
Name	String	To get the source type name.
SourceType	Integer	To get the source type constant.

Straight

Syntax `Straight = [Parent].Straight`

Purpose To define a Straight source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

Table

Syntax	<code>Table = [Parent].Table</code>
Purpose	To define a Table source type.
Attributes	Refer to Class Description (Table) on page 377.
Methods	Refer to Class Description (Table) on page 377.

UntilStop

Syntax	<code>UntilStop = [Parent].UntilStop</code>
Purpose	To define an UntilStop source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

YawAngleConstant

Syntax	<code>YawAngleConstant = [Parent].YawAngleConstant</code>
Purpose	To define a YawAngleConstant source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Constant	Double	To get/set the constant value.
YawAngle	Double	To get/set the yaw angle value
HoldCurrentValue	Boolean	To get/set the hold current value flag.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

Standstill

Purpose

To specify a Standstill profile.

Class Description (Standstill)

Syntax

No direct creation

Purpose

To specify a Standstill profile.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
Name	String	To get the name of the Standstill profile.
Type	LongitudinalProfileTypes ¹⁾	To get the longitudinal profile type.

¹⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

Methods

—

StimulusAngle

Purpose

To specify a StimulusAngle profile.

Class Description (StimulusAngle)

Syntax No direct creation.

Purpose To specify a StimulusAngle profile.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Constant, Ramp, Table, Manual, Extern, Fixed, Step, Pulse, Sine, or SineWithDwell ¹⁾	To get the active element.
AvailableElements	String[]	To get a list of all the available elements.
Name	String	To get the name of the StimulusAngle profile.
Type	LateralProfileTypes ²⁾	To get the type of the lateral profile.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Table](#) on page 370, [Manual](#) on page 365, [Extern](#) on page 362, [Fixed](#) on page 364, [Step](#) on page 368, [Pulse](#) on page 365, [Sine](#) on page 367, or [SineWithDwell](#) on page 367.

²⁾ Refer to [LateralProfileTypes](#) on page 387.

Methods —

StimulusTorque

Purpose To specify a StimulusTorque profile.

Class Description (StimulusTorque)

Syntax No direct creation.

Purpose To specify a StimulusTorque profile.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Constant, Ramp, Table, Manual, or Extern ¹⁾	To get the active element.
AvailableElements	String[]	To get a list of all the available elements.
Name	String	To get the name of the StimulusAngle profile.
Type	LateralProfileTypes ²⁾	To get the type of the lateral profile.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Table](#) on page 370, [Manual](#) on page 365, or [Extern](#) on page 362.

²⁾ Refer to [LateralProfileTypes](#) on page 387.

Methods

—

Stop

Purpose

To specify a Stop profile.

Class Description (Stop)

Syntax

No direct creation

Purpose

To specify a Stop profile.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	StopWithin or UntilStop ¹⁾	To get the active element.
AvailableElements	String[]	To get all available elements
Name	String	To get the name of the Stop profile.
Type	LongitudinalProfileTypes ²⁾	To get the longitudinal profile type.

¹⁾ Refer to [StopWithin](#) on page 369 or [UntilStop](#) on page 370.

²⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

Methods

—

StopWithin

PurposeTo get information about the **StopWithin** longitudinal type.

Class Description (StopWithin)

Syntax

No direct creation

PurposeTo get information about the **StopWithin** longitudinal type.**Attributes**

The class contains the following attributes:

Attributes	Type	Purpose
SourceType	StopWithinSource ¹⁾	To get the StopWithinSource object for defining distance parameters.
Name	String	To get the name of the longitudinal type.
Type	LongitudinalProfileTypes ²⁾	To get the type constant of the longitudinal type.

¹⁾ Refer to [StopWithinSource](#) on page 374.²⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

StopWithinSource

PurposeTo get information about the **StopWithinSource** type.**Where to go from here****Information in this section**

[Class Description \(StopWithinSource\)](#)..... 375
 To describe the class and its attributes.

[Activate](#)..... 375
To activate a StopWithinSource type.

Class Description (StopWithinSource)

Syntax No direct creation

Purpose To get information about the StopWithinSource type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	StopWithin ¹⁾	To get the active StopWithinSource type object.
AvailableElements	Strings (collection)	To get the names of all available StopWithinSource types.

¹⁾ Refer to [StopWithin](#) on page 369.

Methods The class contains the following methods:

Method	Purpose
Activate	To activate a StopWithinSource type. Refer to Activate on page 375.

Activate

Class StopWithinSource

Syntax `StopWithinSourceType = StopWithinSource.Activate(Variant Type)`

Purpose To activate a StopWithinSource type.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific source type. The value can be specified as string (name) or integer (constant type).

Return value

The method returns an object of the following type:

Type	Description
StopWithin ¹⁾	A source type object according to the given parameter.

¹⁾ Refer to [StopWithin](#) on page 369.

Related topics**References**

[Class Description \(StopWithinSource\).....](#) 375

Table

Purpose

To define a Table source type.

Where to go from here**Information in this section**

Class Description (Table).....	377
To describe the class and its attributes.	
Export.....	377
To write the values of a table to a file.	
Import.....	378
To read values of a table from a file.	
Insert.....	379
To insert an entry in a table.	
Item.....	380
To get a specific entry of a table.	
Remove.....	380
To remove an entry of a table.	

Class Description (Table)

Syntax `Table = [Parent].Table`

Purpose To define a Table source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
Count	Integer	To get the number of entries.
InterpretationType	InterpretationTypes ¹⁾	To get/set the interpretation type.
Name	String	To get the source type name.
Type	Integer	To get the source type constant.

¹⁾ Refer to [InterpretationTypes](#) on page 387.

Methods The class contains the following methods:

Method	Purpose
Export	To write the values of a table to a file. Refer to Export on page 377.
Import	To read values of a table from a file. Refer to Import on page 378.
Insert	To insert an entry in a table. Refer to Insert on page 379.
Item	To get a specific entry of a table. Refer to Item on page 380.
Remove	To remove an entry of a table. Refer to Remove on page 380.

Export

Class Table

Syntax `RetVal = Table.Export(String File)`

Purpose To write the values of a table to a file.

Parameters

The method uses the following parameters:

Parameter	Type	Description
File	String	The file name to which the values are written.

Return value

The method returns an object of the following type:

Type	Description
Boolean	True if successful.

Related topics**References**

[Class Description \(Table\)..... 377](#)

Import

Class

Table

Syntax

```
RetVal = Table.Import(String File)
```

Purpose

To read values of a table from a file.

Parameters

The method uses the following parameters:

Parameter	Type	Description
File	String	The name of the file which values are read.

Return value

The method returns an object of the following type:

Type	Description
Boolean	True if successful.

Related topics**References**

[Class Description \(Table\)..... 377](#)

Insert

Class

Table

Syntax

```
TableValue = Table.Insert(double x, double y)
```

Purpose

To insert an entry in a table.

Parameters

The method uses the following parameters:

Parameter	Type	Description
x	Double	The x value.
y	Double	The y value.

Return value

The method returns an object of the following type:

Type	Description
TableValue ¹⁾	The inserted value of the table.

¹⁾ Refer to [TableValue](#) on page 381.

Related topics**References**

[Class Description \(Table\)..... 377](#)

Item

Class Table

Syntax `TableValue = Table.Item(object Variable)`

Purpose To get a specific entry of a table.

Parameters The method uses the following parameters:

Parameter	Type	Description
Variable	object	The entry to be accessed.

Return value The method returns an object of the following type:

Type	Description
TableValue ¹⁾	A specific entry of a table.

¹⁾ Refer to [TableValue](#) on page 381.

Related topics

References

[Class Description \(Table\)..... 377](#)

Remove

Class Table

Syntax `RetVal = Table.Remove(object Item)`

Purpose To remove an entry of a table.

Parameters

The method uses the following parameters:

Parameter	Type	Description
Item	object	Item to be removed from the table.

Return value

The method returns an object of the following type:

Type	Description
Boolean	True if successful.

Related topics**References**

[Class Description \(Table\)..... 377](#)

TableValue

Purpose

To access an entry of a Table source type.

Class Description (TableValue)

Syntax

```
TableValue = Table.Item()
TableValue = Table.Insert()
```

Purpose

To access an entry of a Table source type.

Attributes

The class contains the following attributes:

Attributes	Type	Purpose
X	Double	To get the X value.
Y	Double	To get the Y value.

Methods

—

Related topics**References**[Class Description \(Table\)..... 377](#)

Velocity

PurposeTo get information about the **Velocity** longitudinal type.

Class Description (Velocity)

Syntax

No direct creation

PurposeTo get information about the **Velocity** longitudinal type.**Attributes**

The class contains the following attributes:

Attributes	Type	Purpose
SourceType	VelocitySource ¹⁾	To get the VelocitySource object for defining distance parameters.
DependencyType	DependencyTypes ²⁾	To get the DependencyTypes object.
Name	String	To get the name of the longitudinal type.
Type	LongitudinalProfileTypes ³⁾	To get the type constant of the longitudinal type.

¹⁾ Refer to [VelocitySource](#) on page 383.²⁾ Refer to [DependencyTypes](#) on page 329.³⁾ Refer to [LongitudinalProfileTypes](#) on page 388.

VelocitySource

Purpose To get information about the **VelocitySource** type.

Where to go from here Information in this section

Class Description (VelocitySource)	383
To describe the class and its attributes.	
Activate	384
To activate a velocity source type.	

Class Description (VelocitySource)

Syntax No direct creation

Purpose To get information about the **Velocity** source type.

Attributes The class contains the following attributes:

Attributes	Type	Purpose
ActiveElement	Constant, Ramp, Extern, FinalVelocity, Sine, or Table ¹⁾	To get the active velocity source type object.
AvailableElements	Strings (collection)	To get the names of all available velocity source types.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Extern](#) on page 362, [FinalVelocity](#) on page 364, [Sine](#) on page 367, or [Table](#) on page 370.

Methods The class contains the following methods:

Method	Purpose
Activate	To activate a velocity source type. Refer to Activate on page 384.

Activate

Class VelocitySource

Syntax `VelocitySourceType = VelocitySource.Activate(Variant Type)`

Purpose To activate a velocity source type.

Parameters The method uses the following parameters:

Parameter	Type	Description
Type	Variant	The type of the specific source type. The value can be specified as string (name) or integer (constant type).

Return value The method returns an object of the following type:

Type	Description
Constant, Ramp, Extern, FinalVelocity, Sine, or Table ¹⁾	A source type object according to the given parameter.

¹⁾ Refer to [Constant](#) on page 362, [Ramp](#) on page 366, [Extern](#) on page 362, [FinalVelocity](#) on page 364, [Sine](#) on page 367, or [Table](#) on page 370.

Related topics

References

[Class Description \(VelocitySource\)](#)..... 383

Constants for Scenarios

Where to go from here

Information in this section

[Constants for Scenarios..... 385](#)
You can use predefined constants in the tool automation.

Information in other sections

[Overview of the Classes for Creating Scenarios..... 187](#)
The overview gives you a quick overview of the classes for the Scenario Editor.

[Automating Scenarios in Python..... 180](#)
You can change the parameters of the scenario.

Constants for Scenarios

Introduction

You can use predefined constants in the tool automation.

Constants

The following constants exist to automate the scenario creation.

ApproximationType The following constants are used to specify an approximation type:

Value	Description
Approach = 0	Distance becomes smaller than specified value.
Depart = 1	Distance becomes greater than specified value.

ClutchAndGearTypes The following constants are used to specify the setting of clutch and gear:

Value	Description
OpenClutch = 0	Open clutch.
Driver = 1	Clutch and gear are set by the driver model.
Stimulus = 2	Clutch and gear are stimulated.
ReferenceGear = 3	Reference gear is specified.

ConditionTypes The following constants are used to specify a condition:

Value	Description
Duration = 0	Condition is Duration type.
Immediate = 1	Condition is Immediate type.

Value	Description
Endless = 2	Condition is Endless type.
Trigger = 3	Condition is Trigger type.
Position = 4	Condition is Position type.
DrivenDistance = 5	Condition is DrivenDistance type.
Distance = 6	Condition is Distance type.
Timegap = 7	Condition is Timegap type.
Velocity = 8	Condition is Velocity type.
LateralAcceleration = 9	Condition is LateralAcceleration type.
DistanceRefLine = 10	Condition is Distance (along road reference line) type.

ControlModeTypes The following constants are used to specify the control mode:

Value	Description
Position = 0	The control mode is position.
YawAngle = 1	The control mode is yaw angle.

DependencyTypeConstants The following constants are used to specify a dependency:

Value	Description
Absolute = 0	Absolute
Relative = 1	Relative

DirectionTypes The following constants are used to specify a direction:

Value	Description
Left = 0	Left direction.
Right = 1	Right direction.

DownloadSelectionTypes The following constants are used to select the elements for downloading:

Value	Description
All = 0	Maneuver and traffic is downloaded.
Maneuver = 1	The maneuver is downloaded.
Traffic = 2	Traffic content is downloaded.

DriverTypes The following constants are used to specify the driver type:

Value	Description
Slow = 0	The slow driver is used.
Normal = 1	The normal driver is used.
Fast = 2	The fast driver is used.
Custom = 3	Custom settings are used for the driver.
AdaptToRoad = 4	The driver adapts to the road.

DrivingDirectionConstants The following constants are used to specify a direction of travel:

Value	Description
Direct = 0	Direct direction
Oncoming = 1	Oncoming direction

GlobalPropertiesLaneIndexInterpretationTypes The following constants are used to specify the lane index interpretation type of the global properties:

Value	Description
RelativeToPreferredLane = 0	Relative to the preferred lane
RelativeToCenterLane = 1	Relative to the center lane

HeightModeTypes The following constants are used to specify the height mode:

Value	Description
Offset = 0	Height is specified as an offset to the tire radius.
Relative = 1	Height is specified relatively to the road surface.

InsertTypes The following constants are used to specify an insert type:

Value	Description
Before = 0	Insert type is Before.
After = 1	Insert type is After.

InterpretationTypes The following constants are used to specify how values are interpreted:

Value	Description
Time = 0	Values are interpreted as time in seconds.
Distance = 1	Values are interpreted as distance in meters.
DistanceRoute = 2	Values are interpreted as distance

LateralControllerTypes The following constants are used to specify the lateral controller type:

Value	Description
Controller1 = 1	The controller 1 is used. This controller is recommended for maneuvers with road.
Controller2 = 2	The controller 2 is used. This controller is recommended for maneuvers without road.

LateralProfileTypes The following constants are used to specify the lateral profile:

Value	Description
NotUsed = 0	Lateral profile is not used.
Continue = 2	Lateral profile is continue.
Deviation = 3	Lateral profile is deviation.
LaneSelection = 4	Lateral profile is lane selection.

Value	Description
StimulusAngle = 5	Lateral profile is stimulated by an angle.
StimulusTorque = 6	Lateral profile is stimulated by a torque.
BasicRoad = 7	Lateral profile is specified by a basic road.
FollowRoad = 8	Lateral profile is specified by the road.

LineLengthCalculationType The following constants are used to specify how the line length:

Value	Description
ReferenceLine = 0	The actual length of lines and gaps differs from the specified values depending on the distance to the reference line. The specified values are the lengths on the reference line.
Absolute = 1	The actual length of lines and gaps is equal to the specified values.

LongitudinalProfileTypes The following constants are used to specify the longitudinal profile:

Value	Description
NotUsed = 0	Not used
Continue = 2	Continue the last valid definition
Acceleration = 3	Acceleration profile with absolute values
DistanceMeter = 4	Distance of a fellow vehicle to another fellow vehicle in meters
DistanceTime = 5	Distance of a vehicle to another fellow vehicle in seconds
Position = 6	Position of the fellow vehicle from the road start position
Velocity = 7	Velocity of the fellow vehicle
StopWithin = 8	Stop within a distance
PedalStimulus = 9	Stimulus of a pedal
Standstill = 10	Standstill
LateralAcceleration = 11	Lateral acceleration
Stop = 12	Stop

ReferencedVehicleTypes The following constants are used to specify the reference type of a follow road maneuver:

Value	Description
RelativeToLane = 0	Relative to a lane
RelativeToObject = 1	Relative to an object

ReferencePointTypes The following constants are used to specify a reference point of an object.

Value	Description
Main = 0	Main point (front axle center)
Rear = 1	Rear side center
Front = 2	Front side center
Left = 3	Left side center

Value	Description
Right = 4	Right side center
Nearest = 5	Nearest side center

RelativeObjectType The following constants are used to specify a relative object type:

Value	Description
Vehicle = 0	Object is a vehicle.
Marker = 1	Object is a position marker.

RelativeOperatorType The following constants are used to specify a relative operator:

Value	Description
GreaterThanEqual = 0	Relation is greater than or equal.
LessThanEqual = 1	Relation is less than or equal.

ReplacementTypes The following constants are used to specify the replacement type:

Value	Description
Maneuver = 1	Replaces the maneuver.

SelectorLeverTypes The following constants are used to specify a selector lever type:

Value	Description
Driver = 1	Selector lever is specified by the driver model.
Stimulus = 2	Selector lever is stimulated.

SourceTypeConstants The following constants are used to specify the source type:

Value	Description
Constant = 0	The source type is Constant.
Ramp = 1	The source type is Ramp.
FinalDistance = 2	The source type is FinalDistance.
FinalVelocity = 3	The source type is FinalVelocity.
FinalDeviation = 4	The source type is FinalDeviation.
Sine = 5	The source type is Sine.
SmoothLaneChange = 6	The source type is SmoothLaneChange.
External = 7	The source type is External.
FinalLane = 8	The source type is FinalLane.
LaneChange = 9	The source type is LaneChange.
StopWithin = 10	The source type is StopWithin.
SimpleConstant = 11	The source type is SimpleConstant.
YawAngleConstant = 12	The source type is YawAngleConstant.
Table = 13	The source type is Table.

Value	Description
UntilStop = 14	The source type is UntilStop.
Fixed = 15	The source type is Fixed.
Step = 16	The source type is Step.
Pulse = 17	The source type is Pulse.
SineWithDwell = 18	The source type is SineWithDwell.
Circle = 19	The source type is Circle.
Straight = 20	The source type is Straight.
Manual = 21	The source type is Manual.

SpeedLimitModes The following constants are used to specify the speed limit mode for a traffic driver:

Value	Description
IgnoreScenery = 0	The speed limit specified in the scenery of the road is ignored.
UseScenery = 1	The speed limit specified in the scenery of the road is evaluated.

StopTypes The following constants are used to specify the stop type:

Value	Description
StopWithin = 8	The ASM vehicle stops within a specified value.
BrakeUntilStop = 9	The ASM vehicle brakes until it stops.

Related topics

References

[Overview of the Classes for Creating Scenarios..... 187](#)

Limitations

Limitations of the Scenario Editor

Limitations

Tool automation The tool automation interface of the Scenario Editor has been changed in ModelDesk 4.1. It is not possible to use older scripts with the new Scenario Editor.

A

- acceleration
 - specifying 45
- Acceleration class 323
- AccelerationSource class 324
- activate scenario 94
- ActiveTrafficScenario class 198
- activity 21
 - basics 31
- activity (fellow)
 - properties 119
- activity (global user signal)
 - properties 120
- activity (maneuver)
 - properties 121
- adding
 - segment 43
 - sequence 43
- alias variables
 - scenario 72
- ApproximationOperator 327
- ApproximationOperators 326
- ASM vehicle 21
- automating
 - maneuver control 184
 - scenario 180

B

- braking
 - specifying 45

C

- classes overview
 - Scenario Editor 187
- clutch and gear (MT)
 - properties 122
- combining demo scenarios with roads 38
- Common Program Data folder 18
- ConditionDistance class 283
- ConditionDistanceRefLine class 283
- ConditionDrivenDistance class 284
- ConditionDuration class 285
- ConditionEndless class 286
- ConditionImmediate class 287
- ConditionPosition class 288
- ConditionTimegap class 289
- ConditionTrigger class 290
- ConditionVelocity class 290
- configuration page
 - scenario 94
- controlling
 - driving cycle 76
 - driving cycles 74
 - maneuver 76
 - maneuvers 74

D

- defining

- maneuver user signal 56
- DependencyTypes 329
- DistanceTime class 334
- DistanceTimeSource class 335
- Documents folder 18
- driver model 54
- driving a route 48
- driving cycles
 - controlling 74

F

- fellow 21
 - basics 29
 - creating 62
 - geometry 29
 - initial value 30
 - initializing 63
 - properties 126
 - specifying the movement 63
- fellow definition
 - activity 34
- fellow user signal 30
 - creating 60
 - properties 127
- FellowActivity class 242
- FellowTransition class 255

G

- gear
 - shifting 46
- geometry of a fellow 29
- global driver
 - specifying 54
- global user signal
 - activity 35
 - basics 31
 - properties 129
- GlobalUserSignal class 269
- graphical user interface
 - Scenario Editor 22

L

- lane change maneuver 50
- LaneSelection class 341
- LaneSelectionSource class 342
- lateral type (fellow)
 - properties 129
- lateral type (maneuver)
 - properties 133
- LateralDeviation class 339
- LateralDeviationSource class 339
- Local Program Data folder 18
- longitudinal type (fellow)
 - properties 143
- longitudinal type (maneuver)
 - properties 151

M

- maneuver

- activity 31
 - basics 27
 - initial value 28
 - properties 158
 - reset 175
 - start creation 41
 - steering 52
 - transition 53
- maneuver control
 - workflow 75
- maneuver control automation 184
- maneuver user signal 28
 - defining 56
 - properties 159
- ManeuverControl class 193
- maneuvers
 - controlling 74
- modify scenario 94

P

- Position class 350
- PositionSource class 351
- properties
 - activity (fellow) 119
 - activity (global user signal) 120
 - activity (maneuver) 121
 - clutch and gear (MT) 122
 - fellow 126
 - fellow user signal 127
 - global user signal 129
 - lateral type (fellow) 129
 - lateral type (maneuver) 133
 - longitudinal type (fellow) 143
 - longitudinal type (maneuver) 151
 - maneuver 158
 - maneuver user signal 159
 - segment 160
 - selector lever (AT) 161
 - start (fellow) 163
 - start (global user signal) 164
 - start (maneuver) 164
 - traffic scenario 166
 - transition 167

R

- RelativeObject 355
- RelativeOperators 352, 353
- RelativeVehicleSelection 355
- reset maneuver 175
- road
 - selecting for scenario 41
- route
 - driving 48
- Route 357
- RouteSelection 358

S

- scenario 26
 - activate 94
 - configuration page 94

- exporting 71
- importing 71
- modify 94
- road network 26
- start creation 41
- using 71
- scenario automation 180
- Scenario Editor
 - graphical user interface 22
- scenario name 41
- scenarios
 - combining with roads 38
- segment 21
 - adding 43
 - properties 160
- selector lever (AT)
 - properties 161
- sequence 21
 - adding 43
- shifting
 - gears 46
- specifying
 - acceleration 45
 - braking 45
- start (fellow)
 - properties 163
- start (global user signal)
 - properties 164
- start (maneuver)
 - properties 164
- start segment 21
- steering
 - maneuver 52
- StopWithin class 374
- StopWithinSource class 374

T

- traffic driver
 - new 86
 - open dialog 87
 - specifying properties 69
- traffic ini 71
- traffic scenario
 - properties 166
- transition 21
 - basics 36
 - maneuver 53
 - properties 167

U

- user signal 21
- UserOutput class 264
- UserSignalSegment class 274
- UserSignalSequence class 278
- UserSignalTransition class 279
- UserSignalValue class 280
- using scenarios 71

V

- Velocity class 382

- VelocitySource class 383

W

- workflow
 - maneuver 40
 - maneuver control 75
 - traffic 59