Mechanical Simulation
755 Phoenix Drive, Ann Arbor MI, 48108, USA
Phone: 734 668-2930 • Fax: 734 668-2877 • Email: info@carsim.com

CarSim
carsim.com

CarSim 2020.1 New Features

This document lists notable new features in CarSim version 2020.1.

VS Solver: Architecture

Specifying Model Layout

When running a simulation, previous versions received the model layout (suspension types and trailer information) from the Simfile. This is now done with a VS Command `MODEL_LAYOUT` that is located at the top of the Parsfile and provides a text description of the layout, such as `I_I` for a vehicle with two independent suspensions. Until this command is provided, the VS Solver only recognizes some system-level parameter names. Nothing related to the math model is installed until the MODEL_LAYOUT Command is applied.

In CarSim, another specification is provided by a command SET_DUALS, which enables wheels to support dual tires. This command is still provided in the Simfile when running from the Browser but may also be set with a VS Command located at the top of the Parsfile. If placed in the Parsfile, it must be set before the MODEL_LAYOUT command.

Running Multiple Vehicles in the Solver

The new MODEL_LAYOUT command supports up to four separate vehicle models that can run simultaneously. The text description provided to the MODEL_LAYOUT command should give the layout codes for each vehicle, separated by spaces. For example, the code I_I I_I I_S would setup a simulation with three vehicles, two with front and rear independent suspensions, and the third with a solid-axle rear suspension.

A multi-vehicle simulation contains everything a regular simulation has: system settings for file outputs, time steps, and so on; a ground surface defined with a set of road surfaces or a VS Terrain file; a set of VS paths for use by driver model(s) and for locating moving objects; moving objects that can represent pedestrians, building, signs, etc.; and sensors attached to vehicle units and moving objects. The difference is that there can be up to three additional vehicles included in the same simulation.

A new index parameter IVEHICLE is incremented for each dataset from the CarSim **Vehicle: Assembly** library. Vehicle control systems such as steering, braking, and powertrain have been extended to include separate modules for each vehicle. The same is true with the closed-loop controllers for steering and controlling speed, and the built-in electronic stability controller (ESC).

There are a few limits in the 2020.1 version:

- The engine-mount option may only be used for the first vehicle.
- The frame-twist option may only be used for the first vehicle.
- The legacy 10-point preview model in the driver model may only be used for the first vehicle.
- The hybrid electric powertrain option may only be used for the first vehicle.
- The Preview Points for External Driver Control option may only be used for the first vehicle.
- The option to add a longitudinal degree of freedom to an independent suspension via the DEFINE_SUSP_X_DOF command is not supported if the simulation has multiple vehicles. (Using the command with multiple vehicles will cause the solver to crash.)

The existing library tool named **Multiple Vehicles** was renamed as **Parallel Solvers**, to distinguish its support as an alternative means of simulating multiple vehicles by running multiple solvers in parallel.

VS Commands

User-Defined Functions

User-defined functions no longer require a return variable when called. If a Defined Function has a nominal return that is not needed, it can be called without the return and the return value is ignored.

Indexed Parameters

Indexed parameters (such as `A_KPI(1, 1)`) can now accept variables or equations for their indices. The parsing of indexed parameters is also more robust.

VS Solver: Models

New Options for Attaching ADAS Sensors and Moving Objects

ADAS sensors may now be attached to moving target objects that are not part of the vehicle. These sensors can be used to sense other targets, using the standard sets of ADAS detection variables.

Target objects may now be attached to vehicle sprung masses. These may be detected from sensors in other vehicles, or from sensors attached to moving objects. This supports the use of sensors in targets to detect imminent collision.

Two new dimension parameters were added for each sprung mass that are not used directly by the VS Solver, but which may be helpful when attaching a target object to the sprung mass. There were already two dimensions that were not used by the VS Solver but were handy for scaling 3D shapes in VS Visualizer (there were width and height). The new parameters are overall length and distance from the origin to the front of the 3D shape, and these correspond to parameters for rectangular targets. The result is that when a target object is attached to a sprung mass, the names of the four reference dimensions can be used to match the target dataset to any sprung mass without needing to set the size separately.

The four reference sprung-mass dimensions are `HT_SM`, `LEN_SM`, `WID_SM`, and `LX_F_SM`.

Clutch Timing Parameters

CarSim has two screens for specifying automatic shifting with a clutch. One shows parameters involving lags within the shifting, while the other shows points on a timeline. Most users found the timeline version more convenient to use and in the past decade most shipping examples have used Timeline datasets.

Starting with the 2020.0 release, the VS Solver has used the parameters shown on the older parameter screen regardless of which screen was used. Datasets for the Timeline screen included the Timeline values, along with calculated values for the parameters used in the VS Solver. The behavior was correct, but the interface could be confusing because four of the keywords and values shown on the VS Browser screen did not match the keywords and values displayed in Echo files generated when the datasets were used.

As of the 2020.1 release, the VS Solvers use the Timeline parameters directly, providing a match between information in the Echo file and the values shown on the Timeline screen. Corresponding changes were also made in the two VS Browser screens, as described later. The four old parameters are still calculated and shown in the Echo file as read-only CALC parameter to help confirm that new runs are working as expected.

Speed Controller

The speed controller has been extended to support integral control when driving in reverse, allowing the vehicle to reach and maintain the specified target speed.

Path Detection

The *path detector*, for a given path ID and lateral target ID, fits a cubic polynomial to three target points to create an instantaneous preview path. The cubic polynomial is constructed such that it approximates a clothoidal segment (Euler spiral). The preview path outputs include curvature, rate of change of curvature, and lateral offset of the path relative to the vehicle. The implementation uses a Kalman filter to estimate the curve fit based on noisy target point measurements.

The path detector is a standalone feature which calculates outputs based on the specified path information only; it does not control the vehicle or account for the occlusion of the path by moving objects. In this regard, it can be considered a more advanced version of the traditional preview point.

The path detector may have multiple instances, allowing, for example, the left and right markings of a lane to be detected based only on a reference path and `LTARG` datasets. Lane centering examples demonstrating this are included; they use outputs from the path detector(s) to calculate steering requests via VS Commands.

For more information on the path detector, refer to **Help > Model Extensions and RT > Path Detectors.** This covers the basic operation, technical details, and the new library/screen **Custom Forces and Motion Sensors > Path Detectors for External Driver Control**.

Inclined Springs for Solid Axles

Solid axle suspensions in CarSim now support another spring model featuring parameters for spring length, front-view spring inclination angle, and side-view spring inclination angle. This new behavior is not active by default; activation is accomplished by specifying a spring length greater than zero. For more information, refer to the *Suspension Systems* document, available from the **Help** menu. This feature was added while reviewing the damper inclination option introduced for solid axle vehicles in CarSim and TruckSim versions 8.1. Refer to the *Backwards Compatibility* document, available from **Help > Release Notes**, for discussion on improvements to the inclined damper feature.

Virtual Steering Axis Front Suspensions

The virtual steering axis suspension is an independent front suspension which has no fixed kingpin axis. These suspensions arise when a lower or upper control arm (wishbone) is split into two tension-compression links, resulting in double ball joints at the wheel-side of the control arm. In this case, the steering axis is no longer representable by fixed geometric parameters such as kingpin inclination or caster angle. Instead, the steering of the wheel relative to the sprung mass is about an instantaneous steering axis given by the linkage's geometry.

Beginning with CarSim version 2020.1, the virtual steering axis suspension type is modeled at the system level by having the suspension kinematics tables be entered as 2D tables of jounce and steering rack travel. The virtual steering axis suspension is denoted by model code "Vir", or "V" for short. To avoid confusion with the generic/independent suspension denoted by "Ind" or "I", the virtual steering axis suspension is rarely called an independent suspension in CarSim; the "virtual steering axis" name is preferred.

Two vehicles are included in CarSim 2020.1 which make use of the new model: "B-Class, Sports Car (VSA)" and "D-Class, Sedan (VSA)". Noteworthy is that the front suspension of the sports car was created based on a true double ball joint strut suspension model which was exercised in SuspensionSim. The sedan's virtual steering axis front suspension data was created by one of our math model experts to allow it to perform similarly to the existing independent front suspension "D-Class, Sedan".

The virtual steering axis suspension requires the use of three new main libraries (screens): **Suspension: Virtual Steering Axis System Kinematics**, **Suspension: Virtual Steering Axis Compliance, Springs, and Dampers**, and **Steering: Virtual Steering Axis**. Additional new libraries are also available to input the 2D suspension kinematics tables.

Aside from the expected differences in suspension and steering kinematics, the new virtual steering axis model also has some key differences in available import variables, how it uses the suspension compliance coefficients, and how it calculates the steering rack reaction forces. For more on the CarSim virtual steering axis front suspension, see **Help > Suspension Systems** and **Help > Steering Systems**.

Surface Rolling Resistance Coefficient is a Function of S and L

In past versions, the road parameter `RR_SURF` defined the surface coefficient for tire rolling resistance and was fixed for a given road surface. The coefficient is now defined with a Configurable Function that varies with S and L, as is done with road friction. The GUI was updated to replace the keyword `RR_SURF` with `RR_SURF_CONSTANT`. Tables can now be used for road surfaces where the rolling resistance varies with position.

When using VS Terrain, the rolling resistance coefficient is provided to the model based on global X and Y coordinates.

Consecutive Runs on dSPACE ds1006 Board Without Unloading

With a one-time download of the VS Solver to the dSPACE DS1006 board, the simulation can be executed at least 8000 times consecutively.

VS Browser: Graphic User Interface (GUI)

Find All Unreferenced Datasets

A new option **Find All Unreferenced Datasets** has been added to the VS Browser under the **Tools** menu. This produces a list of every dataset in the current database that has no dataset referencing it. This option can be helpful in maintaining the database and identifying currently unused datasets.

Find All Deprecated Assets

A new option **Find All Deprecated Assets** has been added to the VS Browser under the **Tools** menu. This will produce a list of every dataset in the current database that has a reference to a deprecated animator asset. Deprecated assets reside in the `CarSim_Prog\Resources\Animator\Deprecated_Assets` folder of your installation directory. Selecting this option from the **Tools** menu will also produce a list of missing animator assets and write the results to the log file.

The recommended practice is to perform this step on a database before upgrading it to a newer version of the software. If any deprecated animator assets are identified, they should be corrected prior to the database upgrade.

Re-Write All Runs Menu Item is Renamed

The **Tools** menu command to **Re-Write All Runs** has been renamed to **Re-Write every 'All Parsfile' sent to the Solver or VS Visualizer**. This renaming was done to provide more clarity for what action was being performed: to re-write every Parsfile in the **Runs** library in the database. This includes the `All.Par` files that are used to provide information to the VS Solvers and VS Visualizer.

Clutch Timeline and Parameter Screens

CarSim has two screens for specifying automatic shifting with a clutch. One is named **Control: Clutch Shifting Parameters (Closed Loop)** and the other **Control: Clutch Shifting Timelines (Closed Loop)**. The Timeline screen has generally been more popular, but the VS Solvers used the parameters shown on the Parameter screen and showed those keywords and values in Echo file regardless of which screen was used for a simulation.

In 2020.1, the Solvers were changed to use the Timeline parameters, such that they are now shown in Echo files. The Timeline screen has been changed to use the keywords now recognized by the VS Solver. Further, the Parameter screen has been modified to calculate the Timeline parameters and include the proper keywords in the dataset Parsfile.

The 2020.1 VS Browser is backward compatible with datasets imported from older versions. Be aware that the older Parameters screen is now deprecated and is scheduled to be removed from the browser in version 2021.0.

Tire Screen

Support for the Pacejka 5.2 Magic Formula tire has been deprecated and this feature is now supported by the TASS/Delft external tire model. The **Tire** screen now includes a button to export a .tir format file from the linked parameters. Instructions for connecting to the external tire model are available from the document **Help>Deprecated Items>Internal Pacejka 5.2 Tire.**

Miscellaneous Changes to Support ADAS Scenarios

ADAS Sensors for Range and Tracking screen

The options for attaching the sensor to a part in the vehicle have been extended to include two new options:

1. The current unit sprung mass. This allows the same sensor dataset to be applied to a lead unit or trailer and simplifies support for scenarios with multiple vehicles.
2. The current moving object. This allows the sensor to be attached to a target object, enabling detection of collisions with a target object attached to a vehicle.

Single Moving Object (Custom) screen

The options for specifying the motion of the object have been extended to include attaching the object to the sprung mass of the current vehicle unit. This is done in support of simulations with multiple vehicles, as well as for simulations where the sensor is attached to a moving object with the intent of detecting where a collision occurs with the vehicle.

Events screen

A checkbox and associated yellow field are provided to specify the vehicle number for use in simulations with multiple vehicles running from one VS Solver. A line of text is written at the top of the Event Parsfile to set `IVEHICLE` to the specified vehicle number. If any controls are set on the Events screen (throttle, braking, speed controller, etc.), `IVEHICLE` determines which vehicle is affected.

Links to Procedures datasets

Blue links on Generic group screens can now link to **Procedures** datasets. This greatly enhances setting up simulations with multiple vehicles.

Vehicle with Loads, Sensors, Trailer, etc. screen

Miscellaneous yellow fields have been added to the screen, with one for the lead unit and one for the trailer (if the Trailer checkbox is checked).

Vehicle Assembly screen

The **Vehicle: Assembly** screen used to represent a lead unit now increments an index parameter `IVEHICLE`, in support of simulations with multiple vehicles.

Vehicle and Generic screens

More miscellaneous blue links now allow Event datasets and ADAS Sensor and Moving Object datasets to be linked.

Run Control screen

It is now possible to link to a dataset from the **Generic Data Group (More)** library for the Math Model blue link.

Other Tools: Stacks

VS Software Development Kit (SDK)

Python APIs for `vs_output` and `vs_vehicle` have been added to the SDK. A standalone C example for the `vs_vehicle` API has also been added. The Python API for the solvers has been updated and the old API (`Simulation.py`) has been deprecated.

VS Terrain Utility

The tool has been upgraded to support conversions of OBJ files. It uses materials from the MTL file similarly to the materials specified in an FBX to associate friction and rolling resistance values.

A merge function has been added to consolidate multiple VSTERRAIN files. This is useful if converting multiple lanes generated from the VS Browser to form a road surface representation.

The performance test has been updated to be more precise.

HPC Licensing Logging

For installations using the HPC (High Performance Computing) Licensing option, the HPC Licensing Server application now writes log files to the Windows user directory or Linux /var/log directory.

VehicleSim Dynamics Plugin for Unreal Engine

The Plugin has been updated to Unreal Engine 4.25 and now supports Unreal nDisplay. When used in an nDisplay cluster, any VehicleSim solvers in use will only be initialized and processed on the Master Node. The instances of the Plugin running on nDisplay Cluster Slave Nodes will suppress initialization and processing of the VS solvers, allowing VS vehicles in the scene to be synchronized by nDisplay across the cluster and eliminating the need for VS Solver licenses on all the machines in the cluster (only the Master Node requires a VS solver license). Note that Unreal Engine supports nDisplay only on Windows.

VS Visualizer

VS Visualizer can now reload/refresh simulation data without the need to be closed and then relaunched. If the simulation results have changed (i.e., the simulation was modified and rerun) but VS Visualizer is still open, the user may press the hotkey Ctrl+F5 or select **Reload data** from the VS Visualizer's **File** menu. This will load the new data to the currently active VS Visualizer window.

Documentation

The following Reference Manuals were removed from the **Help** menu and moved to the VS SDK:

1. VS API
2. VS Connect API
3. VS Vehicle
4. VS Output/Table API

The following documents have been added to the **Help** menu:

1. The Design Load Condition in CarSim (Technical Memos sub-menu)
2. Clutch Control (Closed Loop) Screen (Deprecated Items sub-menu)
3. Deprecated Animator Assets (Deprecated Items sub-menu)
4. Euro NCAP VRU Tests (Technical Memos sub-menu)
5. Gravity and Acceleration Fields (Technical Memos sub-menu)
6. Internal Pacejka 5.2 Tire **(**Deprecated Items sub-menu**)**
7. Introduction to CarSim (Guides and Tutorials sub-menu)
8. Path Detectors (Model Extensions and RT sub-menu)
9. Sign Conventions in the Steering System (Technical Memos sub-menu)
10. Simulations with Multiple Vehicles (Technical Memos sub-menu)
11. VI-Grade Integration **(**Real-Time and DS Systems sub-menu**)**
12. VS Commands Summary (Reference Manuals sub-menu)
13. VS SDK: The VehicleSim Software Development Kit (Reference Manuals sub-menu)

The following Reference Manuals have been updated:

14. System Parameters in VS Solvers
15. VS Browser (GUI and Database)
16. VS COM Interface
17. VS Commands
18. VS Solver Programs

The following Screen documents have been updated:

19. ADAS Sensors and Moving Objects
20. Driver Control Screens (Control sub-menu)
21. Hitch
22. Paths and Road Surfaces (Paths, Road Surfaces, and Scenes sub-menu)
23. Preferences (Tools sub-menu)
24. Procedures and Events

25. Powertrain for Electric and Hybrid Electric Vehicles (BHV/HEV)
26. Running with Parallel Solvers (Tools submenu, renamed from “Multiple Vehicles”)
27. Steering Systems
28. Suspension Systems
29. Vehicles
30. Vehicles and Sensor Targets (Animator sub-menu)
31. VS Scene Builder (Paths, Road Surfaces, and Scenes sub-menu)
32. VS Terrain (Paths, Road Surfaces, and Scenes sub-menu)

The following Technical Memos have been updated:

33. Automating Runs with the VS API
34. Extending VS Math Models with VS Commands and the API
35. Initialization in CarSim and TruckSim
36. Making Advanced Procedures with VS Commands
37. Validation of VS Vehicle Models

Database

Deprecated Animator Assets

Animator assets that have been moved to the folder `CarSim_Prog\Resources\Animator\Deprecated_Assets` are marked for deprecation. Users can find these assets using the **Find Deprecated** assets tool option in the VS Browser. After two product releases (1 year), the assets will then be removed from the product installations. There are a few methods to include these assets if upgrading an old database to the current version, described in the Deprecated Animator Assets document.

New Examples

The following subsection titles correspond to categories for new examples that all share the prefix “* CS 2020.1,” e.g., *** CS 2020.1 – Multiple Vehicles**.

Acceleration Field Imports

The CarSim vehicle models include import variables that can be used to manipulate the uniform gravitational field. The 2020.1 release includes examples showing how the field can be modified to represent a road with a grade. A new Tech Memo is included in the **Help** menu.

Collision Detection

For the CarSim 2020.1 release more Automated Driving Assistance Systems (ADAS) testing, in the form of the European New Car Assessment Programme (Euro NCAP) Automatic Emergency

Braking (AEB) Vulnerable Road User (VRU) simulated tests. The CarSim examples that are included from the Euro NCAP AEB VRU suite of tests are the Car-to-Pedestrian Nearside Adult 25% (CPNA – 25) and the Car-to-Pedestrian Nearside Child 50% (CPNC – 50) tests.

Also included are other supporting examples which help the user define exactly where the ego vehicle (Vehicle Under Test or VUT) will strike the pedestrian target. These are in addition to an example representing just one run of the CPNA – 25 test, with extended HUD outputs for tuning purposes. There is a version of the CPNA – 25 test example in CarSim which moves the AEB and forward collision warning (FCW) logic and tuning from VS Commands to an included Simulink model. Finally, there is one CPNA – 25 example which deliberately includes poor AEB tuning choices to show that this will result in a bad score for the CPNA – 25 test.

K&C Tests

A Kinematics and Compliance (K&C) example has been added that places the vehicle in the Design Load Condition. In CarSim, the Design Load Condition is defined to be when the Sprung Mass Coordinate System Origin is at Z = 0 m, the Pitch and Roll Angles = 0 deg, and the wheel center locations match those on the Sprung Mass screen (CarSim) or Suspension Kinematics screens (legacy CarSim option). This Design Load Condition is thus defined to have equal spring loads side-to-side.

The example defines Reference Points to which custom forces are applied. These are needed to constrain the Sprung Mass body and provide some outputs for the animation. VS Commands are used to implement springs that clamp the Sprung Mass in place and move the ground pads to locate the wheel centers at the design height.

A short Event series is used to monitor the test. If certain calculated parameters exceed their tolerances, the simulation ends.

> **Note** For 2020.1, all K&C tests in CarSim, including the examples that have been shipped for many years, use a road coefficient of friction set to a constant value of zero. Previously the road coefficient of friction was set to 0.00001 to avoid potential divide by zero issues in the tire model. These issues are no longer present, so the coefficient of friction dataset has been changed.
>
> If comparisons are made between 2020.1 and 2020.0 the results will be very close, but some small differences will still be visible. To accommodate direct comparisons, the previous road coefficient of friction dataset is still included in the 2020.1 database; it may be removed in a future release.

Path Detector

Four examples have been added to demonstrate various uses of the new Path Detector feature:

- Detecting a single path (*Path Detector Outputs*)
- Detecting multiple paths (*Multiple Path Detections*)

- Using a detected path for closed-loop steering control (*Lane Keeping Assist System* in CarSim/TruckSim, *Target Lean from Detected Path* in BikeSim)
- Detecting an imported target path (*Imported Target Points*).

Steering Sign Conventions

Two examples have been added that demonstrate a steering system configured to have the Rack and Pinion (R&P) mounted behind the front axle.

When the steering rack is behind the axle, the road wheel steer kinematics are defined such that a leftward (positive) rotation of the pinion produces a rightward (negative) translation of the rack. This produces a leftward (positive) steer of the road wheels.

The power steering boost curve is also representative of the rack behind the axle: boost is in the direction produced by the steering input, so positive torsion bar torque produces negative boost force.

A new Tech memo is included that includes more detail. See **Help > Technical Memos > Sign Conventions in the Steering System.**

Virtual Steering Axis

Three examples have been added to demonstrate the Virtual Steering Axis front suspension module. These use the vehicle data discussed in the previous section "Virtual Steering Axis Front Suspensions" (page 4). Two show dynamic behavior of the virtual steering axis sports car:

- *Double Lane Change, Sports Car (VSA)*
- *Turn 1 (Handling Course),*

while the third example compares the standard D-class sedan to its virtual steering axis counterpart:

- *On-Center Steer Test, D-Class (VSA)*.

Multiple Vehicles

Two examples that were provided in 2020.0 had simulations run under Simulink with a city intersection. One had four vehicles interacting with each other as well as stop signs, a pedestrian, and a bicycle. To run this example, you needed an optional parallel license. The other example had two vehicles and can be run with a basic license. The controller logic was updated for the new version.

For 2020.1 the examples are reproduced without Simulink, using the new multiple vehicle option. In order to rerun these examples, a new multiple vehicle license is needed.