

# CarSim 2021.1 New Features

Database Delivery .....	1
VS Browser: Graphic User Interface (GUI) .....	2
Recent Database Window .....	2
Database Builder .....	2
GUI Support for New Solid Axle Features .....	6
Road: Animator Repeated Object Screen .....	6
Tire (External) Screen .....	6
Support FMU with External Tire Model .....	7
New Time and Version Stamps Added to Parsfiles .....	7
VS Math Models .....	8
S-L from X-Y Coordinate Transformation .....	8
New 2D Suspension Tables .....	8
Miscellaneous .....	9
RT Platforms .....	10
VS Visualizer .....	10
Documentation .....	10
Database .....	12
Database Builder .....	12
New and Updated Examples .....	12

This document lists notable new features in CarSim version 2021.1.

## Database Delivery

In past versions of CarSim, the installer application provides an example database with hundreds of examples covering basic and advanced vehicle dynamics, ADAS, VS Visualizer animation scenarios, basic and advanced uses of built-in controllers, examples using Simulink, LabVIEW, and FMI, and other examples that might be of interest to many users. For example, the 2021.0 release included a little over 400 example runs.

Examples were available separately for specialty setups, such as real-time (RT) systems involving target machines, software development kit (SDK), desktop driving simulator (DS), examples for SAE student competitions, and others.

The delivery method has been changed for 2021.1. As with all versions in the past 15 years, the installer program installs a folder `CarSim_Prog` with our complete program stack (`carsim.exe` GUI Browser, VS Solver and other support libraries, VS Visualizer, VS Scene Builder, other utilities) and static resources such as animator 3D assets and Help PDF files. However, the installer no longer installs a default database. Instead, about 650 examples are provided in over 60 Consolidated Parsfile (CPAR) archive files. Once CarSim is installed, end users may build databases at any time, using any combination of the CPAR archives.

The new tools for installing databases are described in the following section. More information about the specific datasets is provided in the last section (Database, page 12).

## VS Browser: Graphic User Interface (GUI)

The new options for building and managing databases are handled by the VS Browser `carsim.exe`, using a few new features.

### Recent Database Window

When opening recent versions of CarSim, a window appears listing database folders that are known for the current version. This window (Figure 1) has been extended for 2021.1, providing access to a new tool: the Database Builder.

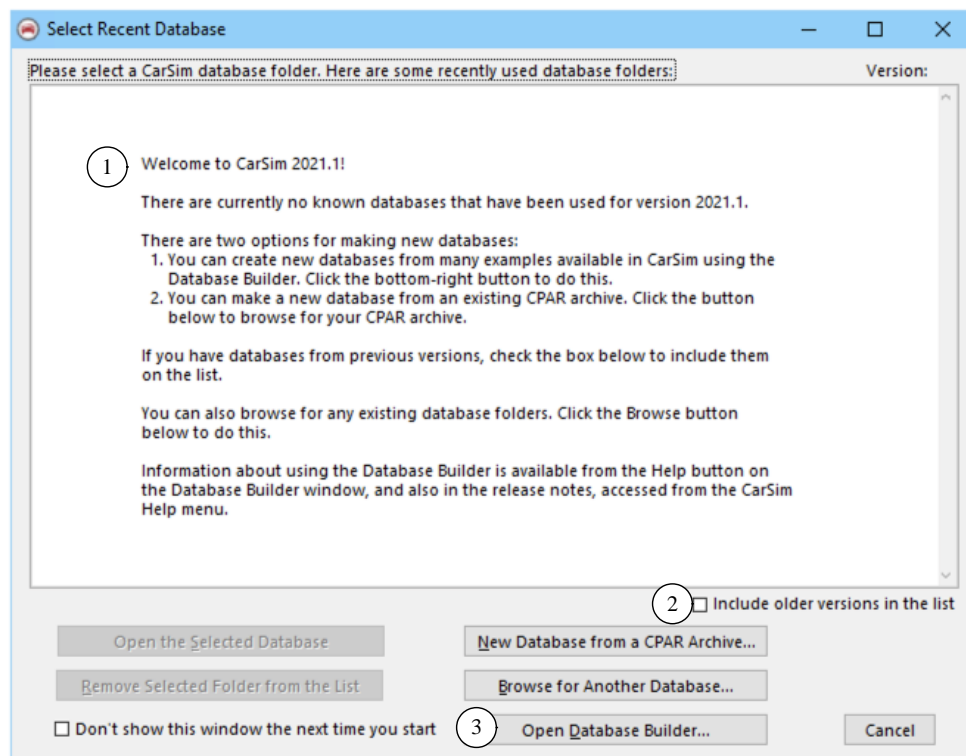


Figure 1. Recent Database window shown when opening a new version of CarSim.

When launching CarSim 2021.1 for the first time, the **Select Recent Database** window shows a message describing options for building a new database or accessing older databases for use in the new version (1). If you have CarSim databases from previous versions, you can check the box **Include older versions in the list** (2) and the window will show databases that were used in the previous version of CarSim. As in past versions, any of those databases may be selected for use in the new version.

### Database Builder

To access examples for 2021.1 when you start CarSim, click the button **Open Database Builder** (3) (Figure 1) to bring up the Database Builder window (Figure 2). When a group of data is selected

①, the button **Build Database from Selected Items** ② becomes active. The **Help** button ③ opens a PDF document that describes the options.

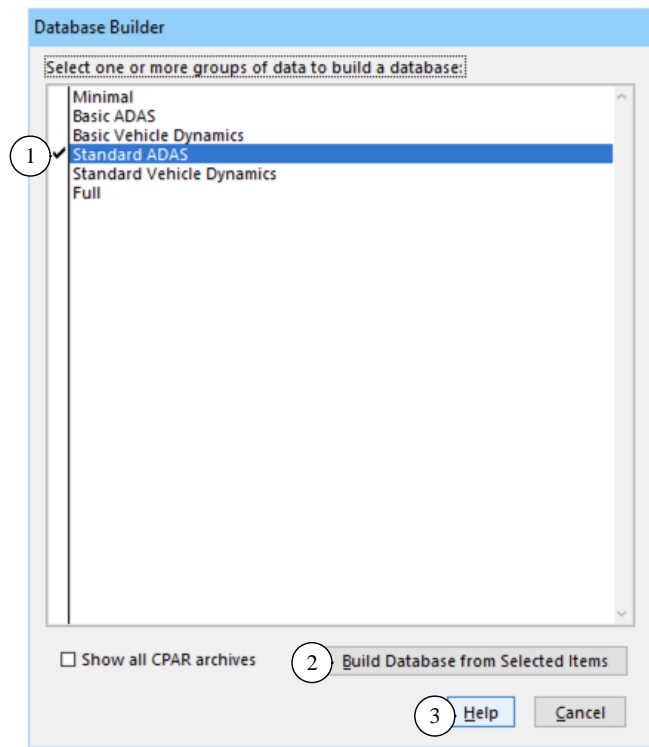


Figure 2. The database builder window showing only predefined database options.

**Note** The Help document is also available from the CarSim **Help** menu in the **Release Notes** submenu.

You can access the Database Builder any time after the initial installation of CarSim. It can be viewed whenever the **Select Recent Database** window appears. The Database Builder can also be brought up directly from within CarSim using the **File** menu ① command **Open Database Builder** ② (Figure 3).

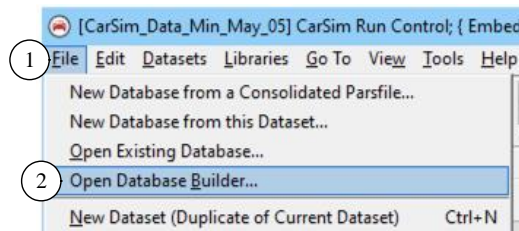


Figure 3. Use the File menu command Open Database Builder.

**Note** Three of the first four options in the **File** menu correspond directly to three of the buttons in the **Select Recent Database** window (Figure 1).

### Predefined database options

Figure 2 shows that CarSim 2021.1 provides six predefined databases. Table 1 provides a little more information about these options.

Table 1. Pre-defined database options.

Database Option	Description	Target Users
Minimal	Examples showing new features, plus core examples such as the Quick Start Guide example, output options, payloads, validation examples, and Preferences	Experienced users with existing databases
Basic ADAS	Minimal + most ADAS examples	New users interested in basic ADAS options
Basic Vehicle Dynamics	Minimal + examples showing vehicle properties and tests	New users interested in basic vehicle dynamics options
Standard ADAS	Basic ADAS + driver model examples, multiple vehicles, roads, proving grounds, LabVIEW, Simulink, VS Visualizer data	Users interested in all examples relevant for ADAS applications
Standard Vehicle Dynamics	Basic Vehicle Dynamics + driver model examples, roads, Proving Grounds, LabVIEW, Simulink, K&C Sequence, vehicle configurations, VS Visualizer data	Users interested in all examples relevant for vehicle dynamics evaluations
Full	Standard ADAS + Standard Vehicle Dynamics + Advanced VS Commands, Custom Forces and Motions, Embedded Python, Extended Models, External Control, Kinematical Preview. Like example databases from older versions.	Users interested in all capabilities that do not require extra licenses or software (other than Sensors, Simulink, or LabVIEW)

### Specialty Examples


None of the six basic options shown in Figure 2 include specialty examples, such as RT HIL systems, software development kit (SDK) examples, examples with trailers or engine mounts or frame twist, examples using external tire models, etc. To access these, and other specific categories of examples, check the box **Show all CPAR archives**  (Figure 4) to see the entire collection. Notice that many CPAR archives are available, shown below the six database options described in Table 1.

Table 2 lists the specialty CPAR archives that are not included in any of the six pre-defined groups. For more information, including the descriptions of all CPAR archives, please see the Help document for the Database Builder.

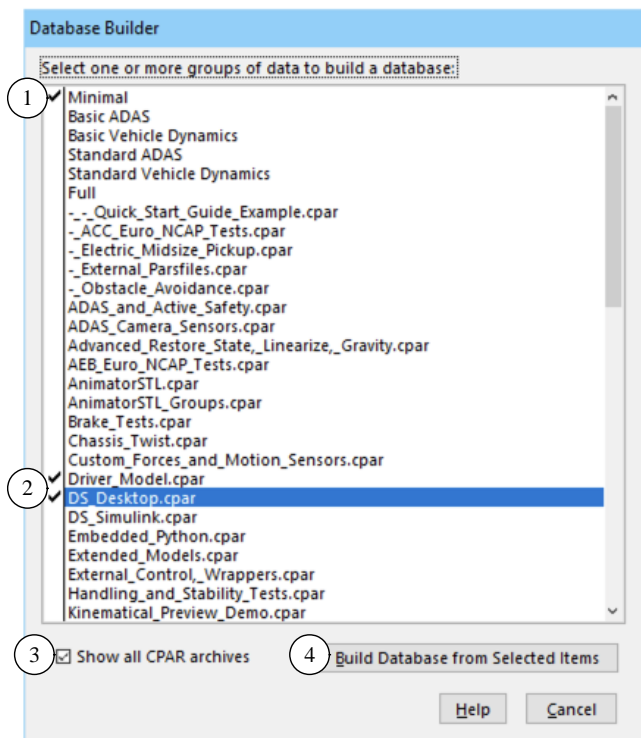


Figure 4. Display when CPARs are also shown.

Table 2. Specialty CPAR archives for standalone use or that require special licenses.

Category	Description
ADAS: Camera Sensors	Use sensor memory shared by VS Visualizer
Chassis Twist	Examples showing the flexible body option
DS: Desktop	Driving simulator examples: stand-alone
DS: Simulink	Driving simulator examples with Simulink
Powertrain (External Software)	Examples with AVL Cruise and GT Suite
Powertrain Mounts	Examples with powertrain mounts
RT: A&D	Examples for A&D RT system
RT: Concurrent	Examples for Concurrent RT system
RT: dSPACE	Examples for dSPACE RT systems
RT: ETAS	Examples for ETAS COSYM and LCO systems
RT: NI	Examples for NI ETS and LinuxRT systems
RT: RT-Lab	Examples for RT Lab system
SAE: name (7 categories)	SAE examples for FSAE and Baja
SDK: Extended Models	Datasets for SDK examples that extend models or provide alternative wrappers
SDK: External Control	
Tire Models (External Software)	Ftire, MF_Swift, TameTire
Trailers (Extra License Needed)	Example using vehicles with trailers
VI-DriveSim Integration	Example for running in VI-DriveSim

More information about specific new dataset categories is provided in the last section of these release notes (page 12).

## GUI Support for New Solid Axle Features

Most of the new features for the solid axle suspension model (described in the following VS Math Model section) are supported in the GUI:

1. Updated **Suspension: Roll Steer (Solid Axle)** library screen, to support 2D table types and combinations of 1D tables.
2. New **Suspension: Dive Angle (Solid Axle)** library screen, to support 2D table types and combinations of 1D tables. The **Suspension: Dive Angle (Caster Change)** library screen is no longer recommended for solid axle suspension types but will still work with legacy datasets.
3. New **Suspension: Longitudinal Movement (Solid Axle)** library screen, to support 2D table types and combinations of 1D tables. The **Suspension: Longitudinal Movement** library screen is no longer recommended for solid axle suspension types but will still work with legacy datasets.
4. Updated **Suspension: Lateral Movement Due to Roll and Jounce** library screen, adding a checkbox for the optional lateral definition enabled by `OPT_SUSP_Y_AXLE_ROLL`. As a reminder, this screen already supported 2D table types and combinations of 1D tables. It is the recommended solid axle lateral movement screen.
5. Updated **Suspension: Solid Axle System Kinematics** screen, to enable linking to the new dive and longitudinal movement libraries. Non-recommended dive, lateral, and longitudinal movement links will be detected and shown as orange instead of blue upon refreshing this screen. Additionally, an empty dataset created using this library screen will now default to the recommended link types for the dive angle, longitudinal movement, lateral movement, and roll steer tables.

## Road: Animator Repeated Object Screen

This screen contained a feature in which objects can be made detectable by sensors. New input fields have been added which allow users to specify the `Lx_Front` and material type of the detected objects.

## Tire (External) Screen

There are some enhancements and cleaning up of the Tire (External) screen to accommodate changes in the external tire models, such as discontinued TNO models that have been replaced by Siemens.

1. Removed **TNO MF-Tyre v6.2** and **TNO MF-Tyre/MF-Swift v6.2** model options as well as all corresponding model solvers, libraries and supporting files for the TNO tire model.
2. Added **Siemens MF-Tyre Only** option which distinguishes all features that can run with a CarSim license on Windows. Any feature item appearing in **Contact**, **Dynamics**, and

- Slip forces** with this MF-Tyre only model option does not require a separate license from Siemens for Windows.
3. Datasets with discontinued model options (**TNO MF-Tyre v6.2** and **TNO MF-Tyre/MF-Swift v6.2**) from prior versions can be imported and automatically assigned to either **Siemens MF-Tyre Only** or **Siemens MF-Tyre/MF-Swift** model options.
  4. External road data file (RDF/CRG format) is now selectable with the Siemens model.
  5. Added **Michelin TameTire** model option which enables you to specify the tire property file (TIR), TameTire model solver for Windows (DLL), RT library (SO file for dSPACE SCALEXIO) and Windows wrapper (VS STI module DLL). The tire property file is automatically converted to a VS format table. The file conversion requires a separate license from Michelin.
  6. The RT Library field can specify not only the library for DS1006 (LIB) but also for Concurrent RT and dSPACE SCALEXIO (SO).

## Support FMU with External Tire Model

The FMU option has been extended to include support for the external tire models MF-Tyre from Siemens and FTire from COSIN. These require that FMU 2.0 be specified.

## New Time and Version Stamps Added to Parsfiles

Parsfiles written by a VS Browser are named using a 36-character unique universal ID (UUID), with a prefix associated with the library, e.g., `Run_6cb07365-85a3-4a3b-95f9-c5c844687601` is a name that might be used for the Run Control library. This is done to support version control software that tracks creation and revisions of files based on their names.

The Parsfiles always include stamps at the end of the file beginning with '#', such as `#Library`, `#Dataset`, `#Category`, `#fileID`, and `#Product`. Three new stamps were added in 2021.1 to provide more information about time and version. Figure 5 compares the content at the end of two similar files made in versions 2021.0 (left side) and 2021.1 (right side). Both are copies of the same file: the Baseline run used for the Quick Start Guide example. The `#FileID` names differ, as expected, because they depend on UUIDs created uniquely for each file. The `#Product` stamps differ, also as expected, because two different versions were used to create and view the files.

Note that the newer version has three new stamps:

1. `#Created` identifies the date and time (obtained from the computer OS) when the file was created. This stamp will never be changed by the Browser even after updates to newer versions.
2. `#Modified` identifies the date and time (obtained from the computer OS) when the file content was last changed. If a change is made, the time-date from the `#Modified` stamp is increased. However, if a change is made and then the Undo command is used (Ctrl+Z or the Edit menu), the file contents are restored, and the time-date value is also restored to the value it had before the change was made.

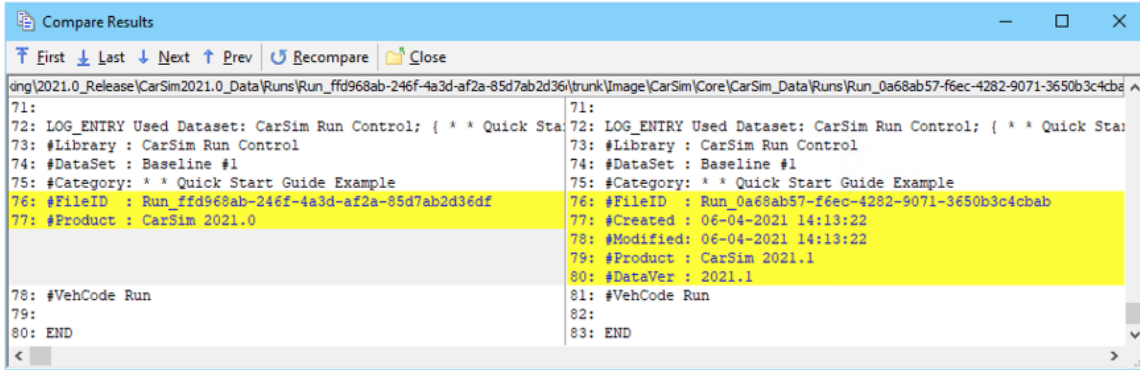


Figure 5. Timestamps compared for versions 2021.0 (left side) and 2021.1 (right side).

- #DataVer identifies the version of the database. With all recent releases, the Database version (#DataVer) has matched the product version (#Product). However, this has not always been the case. (Sometime minor updates have been made for the product, but the database version was not changed.)

## VS Math Models

### S-L from X-Y Coordinate Transformation

Two new options are available for the potentially iterative calculation of S-L (station and lateral coordinate) from X-Y path coordinates during the simulation:

- OPT\_SL\_METHOD (zero or one). When OPT\_SL\_METHOD=1, the estimate for station at any given iteration includes a dependence on path curvature. This can lead to more rapid convergence to a solution for station if the curvature is well-behaved. If the VS Math Model detects a failure to converge, the solution is re-attempted, this time not using path curvature. This is done automatically. When OPT\_SL\_METHOD=0, curvature is never used (the attempt using curvature is bypassed entirely), reproducing the old behavior. The default setting as of 2021.1 is OPT\_SL\_METHOD=1.
- TOL\_SL\_METHOD, to fine-tune the accuracy of the S-L from X-Y iteration. The iteration halts when the distance is less than or equal to TOL\_SL\_METHOD, which has units of millimeters. This existed in the VS Math Model previously but was not settable by the user. The old value was 5 millimeters. The default setting as of 2021.1 is 1 millimeter.

In our testing, the new default settings (OPT\_SL\_METHOD=1, TOL\_SL\_METHOD=1 mm) maintain or slightly improve performance compared to the old behavior (OPT\_SL\_METHOD=0, TOL\_SL\_METHOD=5 mm), while offering improved accuracy.

### New 2D Suspension Tables

Various tables in the suspension models have been updated to support 2D table types.

- The solid axle roll steer, math model keyword SUSP\_AXLE\_ROLL\_STEER, can now be specified with a secondary independent variable of axle jounce. The primary independent variable remains axle roll (roll relative to axle). The 2D effect can be given directly with a



- 2D table type or combined from two 1D tables. The secondary 1D table uses keyword `SUSP_AXLE_JNC_STEER`.
2. The solid axle dive angle, math model keyword `SUSP_DIVE_AXLE`, can now be specified with a secondary independent variable of axle roll. The primary independent variable remains axle jounce. The 2D effect can be given directly with a 2D table type or combined from two 1D tables. The secondary 1D table uses keyword `SUSP_DIVE_AXLE_ROLL`.
  3. The solid axle longitudinal movement, math model keyword `SUSP_X_AXLE`, can now be specified with a secondary independent variable of axle roll. The primary independent variable remains axle jounce. The 2D effect can be given directly with a 2D table type or combined from two 1D tables. The secondary 1D table uses keyword `SUSP_X_AXLE_ROLL`.
  4. The CarSim virtual steering axis suspension now supports 2D tables of jounce and steering rack travel for the jounce and rebound stop kinematics (new VS Math Model keywords `CMP_JSTOP_VIR` and `CMP_RSTOP_VIR`). Previously, these could be entered only as 1D tables of jounce. The additional relationship with rack travel allows the jounce and rebound stops to potentially contribute to the rack force from the tie rod. The two example vehicles in CarSim that use the virtual steering axis — *B-Class Sports Car (VSA)* and *D-Class Sedan (VSA)* — have been updated to include 2D jounce and rebound stop data.
  5. The non-virtual steering axis suspensions now support 2D tables of jounce and jounce other side for the spring and damper compressions (VS Math Model keywords `CMP_SPR_SEAT` and `CMP_DAMP`, respectively). Previously, these only supported 1D functions of jounce. This is most useful for the 2018 twist beam model, which uses the independent suspension with 2D kinematics tables. The example vehicle for the 2018 twist beam model has been updated to use the 2D table type for these.
  6. The non-virtual steering axis suspensions now support 2D tables of jounce and jounce other side for the jounce and rebound stop kinematics (VS Math Model keywords `CMP_JSTOP` and `CMP_RSTOP`, respectively). Previously, these only supported 1D functions of jounce. The example vehicle for the 2018 twist beam model has been updated to use the 2D table type for these.

## Miscellaneous

1. New output variables were added to provide real-time performance: `T_Real_Elapsed` (real clock time since start of run), `T_Real_Step` (real clock time used by the VS Math Model to perform calculations for the last time step, but not counting time used by external software such as Simulink), and `T_Real_Last` (total real clock time used for the last time step by the VS Math Model plus external software).
2. Users can now use VS Commands to define indexed variables and parameters with up to two dimensions. Previously, only indexed variables or parameters of one dimension could be defined by the user.
3. The solid axle lateral movement table, math model keyword `SUSP_Y_AXLE_ROLL`, supports a new option controlled by keyword `OPT_SUSP_Y_AXLE_ROLL`. When `OPT_SUSP_Y_AXLE_ROLL=1`, the solid axle lateral movement table completely

specifies the kinematical lateral displacement of the axle center point relative to the sprung mass. When `OPT_SUSP_Y_AXLE_ROLL=0`, the solid axle lateral movement table contributes partly to the kinematical lateral displacement of the axle center point relative to the sprung mass; there is also a lateral contribution from the axle jounce in the direction implied by axle roll. The setting of `OPT_SUSP_Y_AXLE_ROLL=0` is intended for backwards compatibility with legacy datasets. For new datasets, we recommend turning on the new option using the new checkbox (see notes on new GUI features below).

4. Solid axle suspension types now support a nonzero pitch inertia for the unsprung axle body, math model keyword `IA_YY`. This may be set from a miscellaneous yellow field.
5. Solid axle suspension types now support a nonzero longitudinal CG offset for the unsprung axle body, math model keyword `X_CG_AXLE`. This allows the axle CG's longitudinal location to differ from that implied solely by the wheelbase. This may be set from a miscellaneous yellow field.
6. When the unsprung mass contribution to steering system inertia is turned on (`OPT_I_GEAR_IN=1`), the instantaneous, total steering system inertia value is now printed in Echo files as `I_GEAR_IN_TOT`. The instantaneous inertia value remains available for plotting via the output variable `IstrGear`.
7. A proper error message is added for the case that "Motors on driven axles" is selected with hybrid/electric powertrain but no motor is selected on the differential screen.

## RT Platforms

Starting with version 2021.1, CarSim supports COSYM v2.3 from ETAS with Simulink or FMI 2.0 for COSYM and parallel solvers application.

## VS Visualizer

VS Visualizer can now compute first and second derivative estimates of data channels and use these values for both animation and plotting. See the VS Visualizer reference manual for more information.

## Documentation

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

1. CarSim Database Options (Release Notes)
2. CarSim Known Issues for Converting Old Databases (Release Notes)
3. Euro NCAP ACC Tests (Technical Memos)
4. VS Table Tool Introduction (Reference Manuals)
5. VS Table Tool Usage (Technical Memos)
6. Windows DS for CarSim and TruckSim (Real-Time and DS Systems)

The following Guides and Tutorials have been updated:

7. \* Introduction to CarSim
8. Borrowing a License from a Network Server
9. CarSim Demo Tutorial
10. CarSim Quick Start Guide

The following Deprecated Items have been updated:

11. Clutch Control (Closed Loop) Screen
12. Internal Pacejka 5.2 Tire
13. Path X-Y Coordinate Screens
14. Powertrain Transmission Screens
15. Speed (Closed Loop) vs Station Screen

The following Reference Manuals have been updated:

16. MF-Tyre/MF-Swift User Manual
17. VS Browser (GUI and Database)
18. VS Math Models
19. VS COM Interface
20. VS Commands
21. VS Commands Summary
22. VS Visualizer

The following Screen documents have been updated:

23. ADAS Sensors and Moving Objects
24. Animator Reference Frames
25. Animator Shapes and Groups
26. Animator Wheel Arrows and Other Indicators
27. Batch Matrix
28. CarSim and TruckSim Suspensions
29. CarSim Steering Systems
30. Driver Controls
31. External Models and RT Systems
32. Paths and Road Surfaces
33. Road Surface Visualization

34. Run Control Screen (Home)
35. Tire Models

The following Technical Memos have been updated:

36. Automating Runs with the VS API
37. Convert Simulink Model into Executable
38. Example: Extending a Model with VS Commands and the API
39. Extending a Model with Embedded Python
40. HPC Licensing
41. Parking Lot Layout and Dimensions
42. Simulation with Multiple Vehicles
43. Twist Beam Suspensions: Using 2D Tables
44. VS Solver Wrapper

The following Real-Time and DS System documents have been updated:

45. Concurrent RT Guide
46. NI RT Guide
47. VI Integration Guide

The following Software Development Kit (SDK) documents have been updated:

48. The VehicleSim API — Running and Extending VS Solvers
49. The VS Connect API — Inter-process Data Replication via UDP

## Database

### Database Builder

With the new Database Builder, each category in the Run Control library is provided with a separate CPAR archive file. Many of the categories and titles of existing database were modified. If the content is otherwise the same, then the original UUID file name was kept.

In many of the categories, more runs were included to ensure that older datasets of interest would still be included. The following subsections describe significant new examples or updates of existing examples.

### New and Updated Examples

#### *Solid Axle Examples Use New Recommendations for Library Screens*

The solid axle suspension examples have been updated to use the new **Suspension: Dive Angle (Solid Axle)** and **Suspension: Longitudinal Movement (Solid Axle)** library screens.

### *Euro NCAP ACC Performance Examples*

These CarSim Euro NCAP ACC Performance examples are meant to show users how to implement their own ACC logic and tuning parameters within the CarSim environment, to better understand how their ACC will perform against the Euro NCAP ACC Performance tests. There are 11 examples from the Euro NCAP ACC Test and Assessment Protocol including new Cut-In and Cut-Out examples. Additionally, there are two non-Euro NCAP examples included which are meant to aid in basic ACC testing, before users attempt the more complicated Euro NCAP testing.

### *Online Path Update/Obstacle Avoidance Examples*

This example a passenger car on a path that become obstructed with an obstacle, then uses an onboard autonomous path update procedure to avoid the obstacle, and return to the original path, as a double lane change (DLC). Simulink is used in this example, to create an updated path, and provide updated driver model preview point coordinates, such that the steering controller adjusts the steering to avoid the obstacle. The goal of this example is to show how to change the vehicle path traveled while after the initial path Dataset is loaded. This example also shows how to execute the key steps in the ‘Update Path with Embedded Python’ example, only using Simulink instead of Python and the VS API. There are two examples in this new Obstacle Avoidance Category:

- One utilizes the same spline trajectory as the ‘Update Path with Embedded Python’ example as a double lane change (DLC),
- The other uses a high-order polynomial trajectory as a single lane change (SLC) in path update mode, which then switches back to the CarSim driver model to bring the vehicle back to the original example path.

### *New Midsize Pickup with Fully Electric Powertrain and 2D Solid Axle Kinematics*

The new solid axle features (2D kinematics, pitch inertia, and longitudinal CG offset) are demonstrated with a new fully electric midsize pickup example, including several different test procedures, located in the “\* *Electric Midsize Pickup*” category, part of the minimal database collection. The 2D kinematics data were gathered from a SuspensionSim example of a four-link solid axle suspension with track bar. The electric powertrain is dual-motor, with a 93 kW-h battery pack having a nominal voltage of 723 V.

### *Exotic Sports Car: Powertrain Data Updates*

CarSim ships with two versions of the Exotic Sports Car: one with a 4WD powertrain performing some acceleration tests, and a RWD race car-type version on the Handling Course. The 4WD version received some data updates over the past few releases that had the unintended consequence of preventing it from completing the Event-based Full Throttle Acceleration test.

For the 2021.1 release, this vehicle has received updates in the form of a new value for the driveline frequency (`DRIVELINE_FREQ` is now 9 Hz vs. 12 Hz) and a 7-speed transmission (vs. the previous 6-speed) with new gear ratios and shift schedules. Acceleration and coast-down tests were performed to evaluate the new data over a range of throttle and road grade inputs.

The corresponding Event-based Full Throttle Acceleration test has also been revised to achieve a more realistic launch sequence. The throttle ramp to 100% has been slowed down, as has the clutch release. The brakes, initially applied, are released smoothly. Once the vehicle is rolling and a

specific driveline speed has been reached, Events are used to switch from Open Loop Transmission Control to Closed Loop Transmission Control.

### *External Parsfiles*

The VehicleSim GUI has a screen within the Generic library category called External PARSEFILE. This allows an end-user to link to a parsfile that they create themselves. Once the user-generated parsfile is linked to this screen, the contents of the Parsfile will be provided as input to the VS Math Model if a link is made to the dataset.

In response to questions from many end-users regarding how to link to a parsfile that they have created — i.e., mainly road and path data — the 2021.1 version of CarSim includes two **Run Control** examples that demonstrate the use of the External PARSEFILE library.

### *Updated AEB Example*

The automatic emergency braking example *AEB CCRb w/ Deceleration Command*, which uses the speed controller in target acceleration mode, OPT\_SC=5, to execute the emergency stop, has been updated to prevent conflicting speed controller settings between the vehicle and the procedure.

### *Added More Powertrain Mount Examples*

Five new powertrain mount examples were added, including torque-tube, mount sensors, and hybrid e-AWD.

### *Miscellaneous Examples*

- VS STI simple tire examples are changed to be more robust and showing a proper warning message especially when STI\_TYPARR table is not read.
- VS STI wrapper for FTire model is changed to be more robust and showing a proper warning message especially when an unsupported version of FTire (older than v2020-4) is used.