TruckSim 2021.1 New Features

This document lists notable new features in TruckSim version 2021.1.

Database Delivery

In past versions of TruckSim, 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 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 `TruckSim_Prog` with our complete program stack (`trucksim.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, over 500 examples are provided in 50 Consolidated Parsfile (CPAR) archive files. Once TruckSim 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 `trucksim.exe`, using a few new features.

Recent Database Window

When opening recent versions of TruckSim, 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.
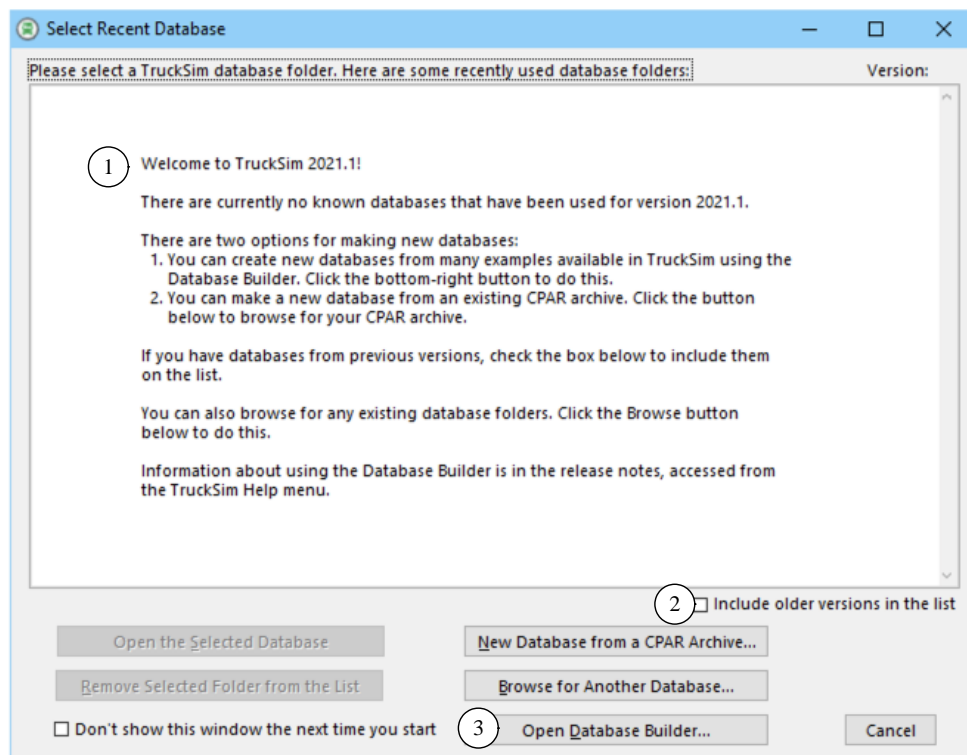

Figure 1. Recent Database window shown when opening a new version of TruckSim.

When launching TruckSim 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 TruckSim 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 TruckSim. 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 TruckSim, 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.
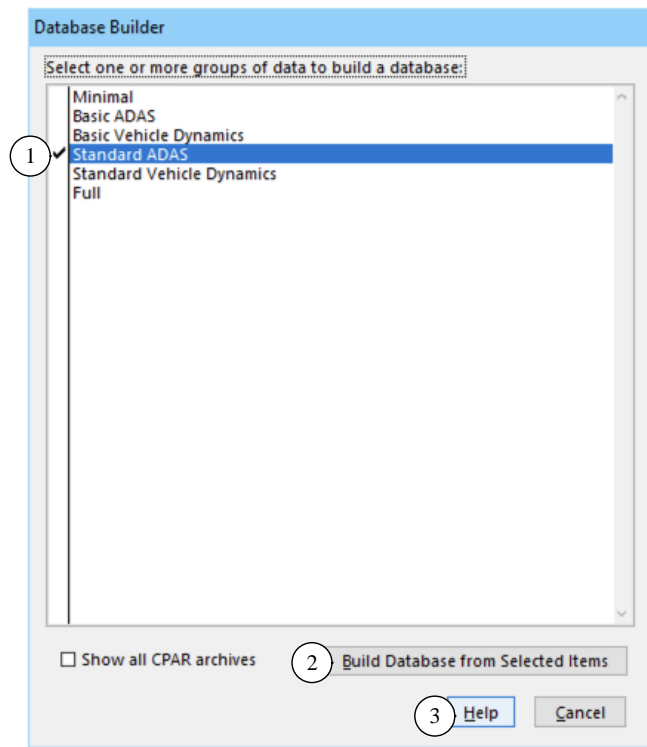

Figure 2. The database builder window showing only predefined database options.

> **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 TruckSim. It can be viewed whenever the **Select Recent Database** window appears. The Database Builder can also be brought up directly from within TruckSim using the **File** menu ① command **Open Database Builder** ② (Figure 3).
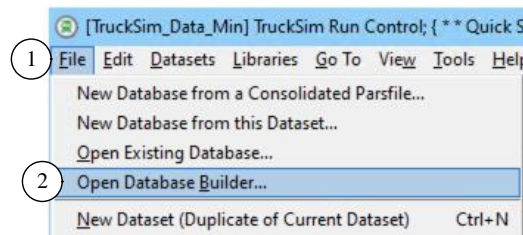

Figure 3. Use the File menu command Open Database Builder.

> **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 TruckSim 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 the Quick Start Guide example, output options, 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, PBS, 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. Like 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 using external tire models, etc. To access these, and other specific categories of examples, check the box **Show all CPAR archives** ③ (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.
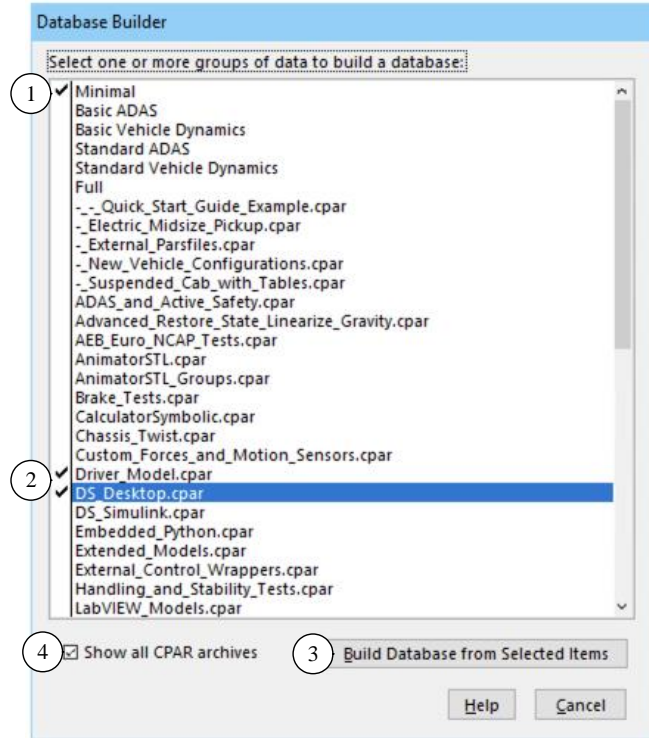
Figure 4. Display when CPARs are also shown.

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

Category	Description
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
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
RT: A&D	Examples for A&D RT system
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

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 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 relating TNO tire model.

2. Added **Siemens MF-Tyre Only** option which distinguishes all features that can run with TruckSim 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. Siemens MF-Tyre/MF-Swift model options are now available for Windows.

5. External road data file (RDF/CRG format) is now selectable with the Siemens model.

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_f02bf0b3-4066-447e-bb5e-40a929b281c7` 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.
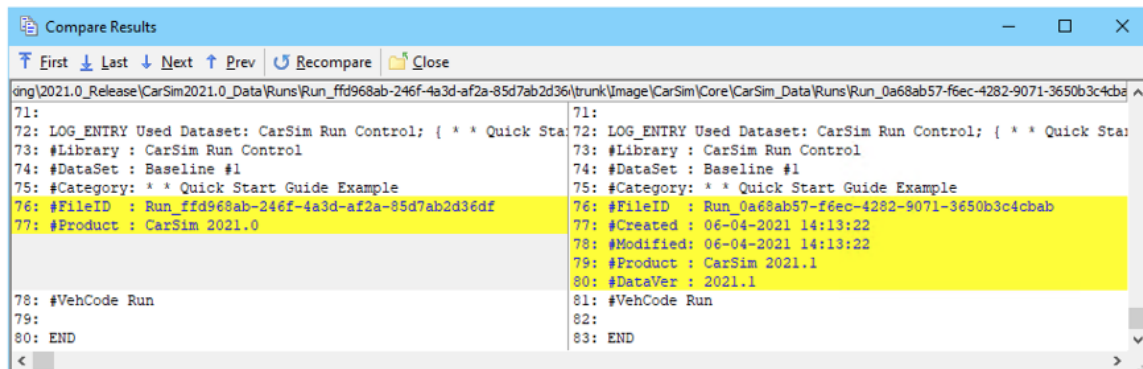

Figure 5. Timestamps compared for versions 2021.0 (left side) and 2021.1 (right side).

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. `#Modfied` 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.

3. `#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:

4. `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`.

5. `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.

1. 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 steered 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.

5. The steered 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), R_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), 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 above).

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, TruckSim 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. TruckSim Database Options (Release Notes)

2. TruckSim Known Issues for Converting Old Databases (Release Notes)

3. VS Table Tool Introduction (Reference Manuals)

4. VS Table Tool Usage (Technical Memos)

5. Windows DS for CarSim and TruckSim (Real-Time and DS Systems)

The following Guides and Tutorials have been updated:

6. * Introduction to CarSim (and TruckSim)

7. Borrowing a License from a Network Server

8. TruckSim Demo Tutorial

9. TruckSim Quick Start Guide

The following Deprecated Items have been updated:

10. Clutch Control (Closed Loop) Screen

11. Internal Pacejka 5.2 Tire

12. Path X-Y Coordinate Screens

13. Powertrain Transmission Screens

14. Speed (Closed Loop) vs Station Screen

The following Reference Manuals have been updated:

15. MF-Tyre/MF-Swift User Manual

16. VS Browser (GUI and Database)

17. VS Math Models

18. VS COM Interface

19. VS Commands

20. VS Commands Summary

21. VS Visualizer

The following Screen documents have been updated:

22. ADAS Sensors and Moving Objects

23. Animator Reference Frames

24. Animator Shapes and Groups

25. Animator Wheel Arrows and Other Indicators

26. Batch Matrix

27. CarSim and TruckSim Suspensions

28. Driver Controls

29. External Models and RT Systems

30. Paths and Road Surfaces

31. Road Surface Visualization

32. Run Control Screen (Home)

33. Tire Models

34. TruckSim Steering Systems

The following Technical Memos have been updated:

35. Automating Runs with the VS API

36. Convert Simulink Model into Executable

37. Example: Extending a Model with VS Commands and the API

38. Extending a Model with Embedded Python

39. HPC Licensing

40. Parking Lot Layout and Dimensions

41. Simulation with Multiple Vehicles

42. Twist Beam Suspensions: Using 2D Tables

43. VS Solver Wrapper

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

44. Concurrent RT Guide

45. NI RT Guide

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

46. The VehicleSim API — Running and Extending VS Solvers

47. 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**.**

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.

External Parsfiles
The VehicleSim GUI has a screen within the Generic library category called External PARSFILE. 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 TruckSim includes two **Run Control** examples that demonstrate the use of the External PARSFILE library.

New Vehicle Configurations

TruckSim supports a wide variety of vehicle configurations ranging from small delivery vans to long road trains with many trailers and dollies. For TruckSim 2021.1, seven new vehicle configuration examples are included that represent 2-axle lead units with 2- or 3-axle trailers, as well as 3-axle lead units with one or two trailers (each with one, two, or three axles), and dollies with one, two, or three axles.

Suspended Cab with Generic Tables for Springs and Dampers

As part of the Frame Torsional Flexibility option, TruckSim supports a suspended cab on the lead vehicle unit with four springs and four dampers. Although the suspended cab GUI screen only has space to represent these springs and dampers as linear coefficients, the TruckSim math model treats these as tables, allowing end-users to represent the cab suspension with nonlinear properties. Import variables for the spring and damper forces further extend the capabilities of the software by allowing detailed representations of the suspension with VS Commands, Simulink, or RT HIL systems.

TruckSim 2021.1 includes four new suspended cab examples that demonstrate how nonlinear springs and dampers this can be set using Generic Tables. Two of the examples demonstrate the springs while two others demonstrate the dampers. These are further broken up by type: Generic Tables with the keyword `imountrow`, and Generic Tables with import variables.

A corresponding set of VS Commands are used to take the data from the tables and apply it to the math model via the appropriate import variables. Although not demonstrated here, it is also possible to have an example in which both the springs and dampers are replaced with Generic Tables and import variables.

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.