

# TruckSim Backward Compatibility

Summary of Major Changes in TruckSim.....	1
Known Long-Term Issues .....	7
Versions 8.2a and Older .....	8
TruckSim 2022.1 vs TruckSim 2022.0.....	8
TruckSim 2022.0 vs TruckSim 2021.1.....	11
TruckSim 2021.1 vs TruckSim 2021.0.....	15
TruckSim 2021.0 vs TruckSim 2020.1.....	18
TruckSim 2020.1 vs TruckSim 2020.0.....	23
TruckSim 2020.0 vs TruckSim 2019.1.....	30
TruckSim 2019.1 vs TruckSim 2019.0.....	32
TruckSim 2019.0 vs TruckSim 2018.1.....	39
TruckSim 2018.1 vs TruckSim 2018.0.....	45
TruckSim 2018.0 vs TruckSim 2017.1.....	50
TruckSim 2017.1 vs TruckSim 2017.0.....	53
TruckSim 2017.0 vs TruckSim 2016.2.....	55
TruckSim 2016.2 vs TruckSim 2016.1.....	61
TruckSim 2016.1 vs TruckSim 2016.0.....	63
TruckSim 2016.0 vs TruckSim 9.0.2.....	66

This document provides a rolling record of bug fixes and backward compatibility issues occurring at each release, going back to 2015 (TruckSim 9.0.2).

There are two companion documents that are accessible from the TruckSim Help menu: **Help > Release Notes > Database Upgrade Guidelines**, which provides recommendations for archiving databases and confirming that upgrading to a new version was successful.

Supported versions of TruckSim are available for download from [www.carsim.com](http://www.carsim.com). If a version is not available for download, then it is not supported. At this time (June 2022), supported versions go back to 9.0. A schedule for the planned end of support is also available on the web site at [www.carsim.com/contactus/endofsupport.php](http://www.carsim.com/contactus/endofsupport.php). The schedule is based roughly on stopping support for software seven years after the initial release and is consistent other engineering software for Windows.

## Summary of Major Changes in TruckSim

Most of this document describes version to version changes in TruckSim. However, some changes have occurred continuously over the past few years and involve multiple versions. These changes are relevant when upgrading from multiple older versions and are described in this section.

## Improvements to Error Handling

On-going improvements to TruckSim have resulted in more stringent error handling. Although the overall effect is improved performance, some fringe-cases that were previously not detected are now being caught, resulting in error messages when the simulation is executed. Examples include:

1. Detailed error checking in VS Commands, including invalid syntax, extra spaces, unbalanced parentheses, and trying to redefine built-in read-only parameters such as ZERO, PI, DR, etc.
2. Out-of-range integer parameters (typically starting with the prefix OPT\_, e.g., OPT\_BK\_SC).
3. User-specified ID parameters < 999 for roads, paths, and LTARG.
4. Tighter tolerances on the looping requirements for paths (coordinates of start and end points must match).

## Modularity of the VS Math Model

Starting with version 9.0 (2015), many parts of the VS Math Model that were originally “built-in” were redefined as modules, in support of many more combinations of features.

### *Payloads, Sensors, Reference Points, and User-Defined Forces*

Prior to version 9, TruckSim included four built-in payloads for each sprung mass. There were ten reference points with associated trios of forces for each sprung mass, and six motion sensors.

These features are now options that can be installed as needed. The built-in parameters for the older versions were discontinued, so old examples using them need to be redone using the current library screens.

### *Paths and roads*

Prior to version 9, the VS Math Model included a single road and two paths (one for the road, one for the driver model). Version 9 provided support of multiple roads and paths. The number of paths and roads were increased in later releases to the current capabilities of 200 roads and 500 paths. Roads and paths have associated user ID parameters that support the re-use of datasets in multiple examples.

If old simulation setups exist in which a road or path is redefined many times with VS Event datasets that in turn link to road or path datasets, it is possible to reach the limit of roads or paths. The solution for this is to use the option in the VS Browser to always set user IDs for paths and roads. This way, repeated uses of a road or path dataset will re-use the existing road or path, rather than always trying to add new ones.

### *Controllers*

Controllers such as ABS and speed control used to be built-in but are now optional (to avoid conflict with external controllers). They are activated by linking to an appropriate screen in the VS Browser or by typing the installation command in a miscellaneous field.

### *The Multibody Math Model*

TruckSim 2018.0 introduced a fully modular architecture for the VS Math Model. All trailer and suspension combinations supported in older versions with custom VS Solver libraries are now handled with just two libraries: `trucksim_32.dll` and `trucksim_64.dll`. Old examples that reference a VS Solver by name must be modified to use the single file names. For example, the old solver for a vehicle model with solid axle front and rear suspensions was `s_s.dll`; this file no longer exists (instead use `trucksim_32.dll` or `trucksim_64.dll`).

Examples affected by this change use the VS API, FMI/FMU, and Simulink examples with copies of a VS Solver DLL file to handle multiple vehicles.

### *Powertrain*

Powertrain modularization was introduced in version 2019.0. This means a two-wheel drive powertrain does not include the drive components for other axles or a transfer case. Simulations involving external software (e.g., Simulink) or VS Commands that use import and export variables that only exist with specific powertrain configurations might have worked in past versions of TruckSim but will generate error messages in newer versions of the software.

Examples that no longer work include runs that change the powertrain using Events, and Simulink models that include I/O Channels for AWD that might have been used with FWD or RWD vehicle models.

## **Discontinued and Deprecated Library Screens**

As the product evolves, so to do the requirements for certain screens. In some cases, the screens were removed. Table 1 lists library screens that have been discontinued since version 8.0.

*Table 1. Library screens that have been removed since version 8.0.*

<b>Library Name</b>	<b>Removed</b>
Cab Custom Forces and Reference Points (Deprecated)	2022.1
Cab Positions, Velocities, and Accelerations (Deprecated)	
Calculator: Original	2016.0
Control: Clutch Shifting Parameters (Closed Loop, Deprecated)	2021.1
Control: Speed (Closed Loop) vs. Station	2021.0
Ground: Elevation Map, X-Y Grid	2017.0
Ground: Friction Map, X-Y Grid	
Models: Transfer to RT-Lab Target v7.x-v8.x	2018.0
Path: X-Y Coordinates	2021.0
Powertrain: Transmission (7 Gears)	
Powertrain: Lockup Schedule	
Powertrain: Unlock Schedule	
Plot: Data Transform	2017.0
Spectrum Analyzer	9.0
Suspensions: Independent System (Legacy Simple)	2019.1

Table 1. Library screens that have been removed since version 8.0.

Library Name	Removed
Suspensions: Solid Axle System (Legacy Simple)	
Tire: Pacejka 5.2 Parameters (Deprecated)	2021.1

Starting with version 2020.0, library screens that are targeted for removal in future versions are formally identified as “deprecated.” Links to datasets in deprecated libraries are shown in orange rather than blue as is done with normal links. The titles are revised to include the word “Deprecated,” and there is usually red text on the screen indicating the library will be removed in a future version. Table 2 lists libraries that are deprecated, with the future version in which the library screen will be removed.

Table 2. Library screens that are deprecated for future removal.

Library Name	Removal
Powertrain: Transmission (External, Deprecated)	2023.0
Powertrain: Downshift Schedule (Deprecated)	
Powertrain: Upshift Schedule (Deprecated)	

The **Help** menu has a submenu **Help > Deprecated Items** (Figure 1) with information about assets that have been deprecated or removed recently. It includes guidelines for moving data from deprecated screens (Table 1 and Table 2), and a document *Deprecated Animator Assets*, described in the next subsection.

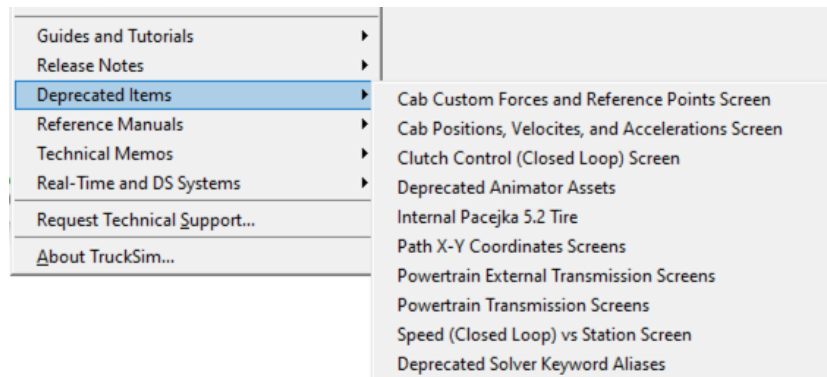


Figure 1. The Help > Deprecated Items submenu.

## Deprecated Animator Assets

To take advantage of modern computer graphics tools and techniques, new 3D shape files are being added that represent a variety of content, ranging from vehicles and tires to environmental details like trees, park benches, and structures. As these new shapes are added, old content is removed. The *Deprecated Animator Assets* document (Figure 1) provides information describing how to identify deprecated shape files and replace them with new shapes. For end-users who have a need

to use deprecated animator assets, the possibility exists to add a `Deprecated_Assets` directory; this will restore animations that might not work if an older database is run with a newer version.

## Ranges for Custom ID Numbers for Paths, Roads, and Tables

Paths, roads, and some tables have user ID parameters to identify the table independently of when it was defined. In version 2017.0, the range of valid ID integers was standardized to either match the internal number (e.g., `PATH_ID(3)` could be set to 3 or any integer number greater than or equal to 999. Setting the ID to a number less than 999 generates an error unless it matches the internal number. For example, attempting to set `PATH_ID(3)` to 7 generates an error.

User-define ID parameters with range limits include: `LTARG_ID`, `PATH_ID`, `ROAD_DZ_ID`, `ROAD_ID`, `SPEED_TARGET_ID`, and `XY_SEGMENT_ID`.

## Using Events to Reload Road Datasets

In general, we do not recommend loading a road dataset more than once in a simulation. When multiple roads are used for different stages of the simulation, we recommend loading all roads initially, and then using VS Events to change the current road using the vehicle parameter `CURRENT_ROAD_ID`. This is sometimes necessary when running with lengthy road descriptions on real-time (RT) systems, where the time needed to load road data might cause the simulation to drop out of hard real time.

The **Road: 3D Surface (All Properties)** library screen has an option to define new datasets using a custom ID. When this option is used, the same road dataset can be used in multiple simulations, with VS Events and other VS Commands referencing it using the custom ID number. When the option is not used, the VS Math Model creates a new road dataset and automatically assigns the current internal number to the ID. For example, the third road created would be automatically given an ID of 3.

Sometimes datasets from older versions of TruckSim that had only a single road (prior to version 9) are imported into newer versions. In those cases, there are two kinds of problems that can occur if any road dataset is read multiple times for a simulation run.

1. If the **Road 3D** dataset does not have custom ID number, a new road is created internally each time the dataset is read. This is OK, unless the VS Math Model reaches the limit in the number of roads allowed. If the limit is reached, an error message is generated, and the simulation will stop.
2. If the **Road 3D** dataset sets a custom ID number, the data updates a road that already exists. This is OK, unless the **Road 3D** dataset has links to one or more dZ datasets that have nonzero incremental elevation. In this case, a new dZ dataset is added to the road each time the **Road 3D** dataset is loaded.

A change was made to the **Road: 3D Surface (All Properties)** library screen in version 2022.1 that resets the number of dZ datasets in the road to zero whenever a custom ID number is used. This eliminates problem 2 above for versions 2022.1 and newer.

## Encrypted Parsfiles

Mechanical Simulation does not support backward compatibility for encrypted expanded Parsfiles. Encrypted data should only be used within the same version of the VS Browser that was used to make the encrypted file (i.e., an encrypted expanded Parsfile exported from TruckSim 2022.0 should only be used with 2022.0 databases).

For versions of TruckSim up to 2021.1, the VS Browser did not always check for older encrypted Parsfiles and sometimes allowed them to be included in databases for newer versions. In these cases, the simulation results are not reliable. Starting with 2021.1, the VS Browser checks properly for encrypted expanded Parsfiles from older versions when importing CPAR archives or updating a database.

At times the encrypted parsfile functionality did not work as intended. Table 3 summarizes the affected versions.

*Table 3. Versions affected by encrypted parsfile issues.*

Version(s)	Functional	Not Functional
2018.1		X
2019.0, .1	X	
2020.0		X
2020.1	X	
2021.0		X
2021.1	X	
2022.0, .1	X	

## No SurfAnim or WinEP

Versions prior to 2017.1 supported older tools for viewing videos and plots: SurfAnim showed animations and WinEP showed plots. The **Preferences** screen had checkboxes for using the older tools: **Use WinEP for plotting** and **Use SurfAnim for video**.

If you get an error message involving SurfAnim or WinEP when you try to view video or plots, go to **Tools > Preferences** and delete any **Preferences** datasets that mention SurfAnim and/or WinEP.

## Discontinuation of ADAS RP

ADAS RP binaries and documentation have been removed from the installation for version 2021.1 and all future releases. ADAS RP binaries can be obtained from versions prior to 2021.1.

## Discontinuation of Direct Connection to AVL Cruise

Prior to the support of FMI/FMI, connection between TruckSim and AVL Cruise was made using a custom interface in the Browser. The interface was removed in 2022.1. Older versions have the interface, but it has not been in use since version 9.

## Third-party Tire Models Settings

The file name and location of the external tire module and external tire solver has been changed. The VS STI Module dll which used to be under `_Prog\Resources\Extensions\External_Tire_API`, and external tire solvers dll should be under the same location as VS solver: `_Prog\Programs\solvers`. The extensions for those file names must be `dll32` and `dll64` for 32 and 64-bit system, respectively.

## Known Long-Term Issues

There are a few issues involving VS Simulation tools that are due to factors outside the control of Mechanical Simulation.

### Results Can Differ Slightly with CPU Architecture

All calculations with floating-point variables are performed in VS Math Model using double-precision variables, which provide significant results out to 15 or 16 significant digits. The precision is far beyond the requirements for providing vehicle dynamic predictions that match measurable test results.

However, there are some small effects that can affect the agreement obtained when comparing results from two versions of a VS Math Model when running on computers with different CPU architecture. Although these effects are too small to be noticeable for most applications of the software, they might cause confusion for activities involving large numbers of simulations made with different computers with different environments (e.g., 32-bit or 64-bit, AMD or Intel).

#### *Differences for 32-bit and 64-bit Windows*

A 64-bit compiler bug with the built-in `atan()` function has been identified that may cause slightly different results between runs in 32-bit and 64-bit Windows solvers. The difference is only one Unit of Least Precision (ULP), and only affects some inputs, but the differences can compound over the course of a run.

The issue can be detected when comparing the output of a run in 32-bit versus 64-bit mode on Windows for long runs.

There is no current workaround. This compiler bug was noticed with internal testing in version 2020.0 and has been filed and accepted by Microsoft. It's likely that this compiler bug existed before 2020.

#### *Differences for Intel and AMD CPUs*

Internal testing at Mechanical Simulation has shown that minor differences in simulation outputs exist for simulations that are made with Windows machines using AMD CPUs as opposed to those with Intel CPUs. The differences are tiny but might cause confusion in automated testing with multiple computers if the computers do not all have the same architecture.

### Performance Loss Using COM and MATLAB/Simulink

A performance regression was identified when using the VS Browser, Windows COM, and MATLAB/Simulink 2019b. Mechanical Simulation's testing results demonstrate a significant

computational performance loss when co-simulating with MATLAB/Simulink 2019b and newer. Mechanical Simulation has notified MathWorks of this issue. Mechanical Simulation can provide a workaround to VehicleSim users upon request which mitigates the performance regression. Please contact our support center (<https://www.carsim.com/support>) for additional information concerning this workaround. For questions about operation of MATLAB/Simulink please contact MathWorks.

## Versions 8.2a and Older

As noted earlier, versions prior to 9.0 are no longer supported. Even so, many simulations defined in versions 8.0 and later will work as before when imported to the newest version. Be aware that some will fail due to major changes made in the software in version 9.0 that are simply incompatible with older versions. If starting the upgrade process with version 8.2a or older, we recommend first updating to a version between 9.0.2 and 2020.0 before upgrading to a later version. These versions include four documents in **Help > Release Notes** about Version 9 Backward Compatibility that provide more information about differences in architecture between versions 8.2.2 and older and versions 9.0 and newer.

## TruckSim 2022.1 vs TruckSim 2022.0

### Bug Fixes

The following bugs were identified and fixed in this release.

#### *VS Math Model*

- The output variable  $J_{ncA\_Ai}$  where  $i$  is an axle number is supposed to be the vertical acceleration of the axle coordinate system origin. In recent version (2018.0 – 2022.0) it was set to zero, due to a change in the VS Math Model architecture that was not noticed at the time.
- The calculations of the hybrid electric and battery electric vehicle (HEV/BEV) powertrain model did not consider that the required power might exceed the battery discharge limit, in which case the motor torque would be limited. However, with the current sequence of calculations, there can be conditions where the calculated energy consumption for the battery is limited due to the discharge limit, but the motor torque was not limited. As a result, the battery had a lower energy consumption for the applied motor torque.

This discrepancy has been fixed in the VS Math Models by having the driving motor torques be regulated by discharge power limit of battery circuit. Also, the regulation of regenerative motor torques is refined by including motor efficiency.

- When using two electric motors on a single axle (select the drop-down menu item “Two electric motors, one on each side” that is found on the differential screen), each motor is directly coupled to a wheel via a half-shaft which does not go through a differential gear. However, the inertia of the electric motor was affected by the differential gear. This discrepancy has been fixed so that the differential gear has no effect to the electric motor inertia with two electric motors on one axle.



- When using two electric motors on a single axle (select the drop-down menu item “Two electric motors, one on each side” that is found on the differential screen), left and right averaged motor inertia is added on each wheel although those two motors are independent each other. This discrepancy has been fixed so that the left and right motor inertia is added on each side of wheels, respectively.

### *VS Browser*

- An issue was discovered involving the import of old databases (prior to version 9) made for VS Math Models that only had a single road that involved reading datasets from the **Road: 3D Surface (All Properties)** library multiple times, typically using VS Events. The problem was described earlier (section *Using Events to Reload Road Datasets*, page 5). A problem that could occur if a road with a user ID number was loaded more than once was fixed in this release.
- When importing old dataset which involves TNO Delft-Tyre with 2D enveloping road contact model, `ISWITCH` value was not updated correctly, and an error is likely to occur when the simulation runs. This import error has been fixed so that the old 2D enveloping contact model is automatically assigned to 3D enveloping contact model with the latest Siemens Tire model and `ISWITCH` is correctly updated.

### *VS Visualizer*

When multiple instances of VS Visualizer were running, attempting to close them all from Windows File Explorer would sometimes fail. The management of file status by VS Visualizer was modified to avoid this issue.

### *VS Scene Builder*

- Scenes that were created with OpenDRIVE scene tiles that were imported by the user failed to generate the required road FBX file in the correct directory. As a result, the road was not displayed in VS Visualizer.
- Resolve issue when deleting a Tile from the Tile directory while the tile is in the scene.
- Improved application stability when importing large files in 32-bit version.

### *VS Terrain Utility*

- Improved application stability when importing large files in 32-bit version.

## **Backward Compatibility**

### *Change to tables for setting open-loop shifting*

The Browser screens for setting transmission gears or maximum gear in use allowed the table type to include interpolation options. The options are now properly limited to constant, step table, or equation.

### *Changes to names of output variables for tow bars and articulation systems*

Version 2022.1 adds output variables for every trailing unit for the global coordinates of a point fixed in the trailing unit sprung mass, located at the rotation point for the front hitch. In the case of a generic or pintle/ball hitch, the hitch point is in the sprung mass coordinate system with local coordinate (0, 0, H\_H\_FRONT), where H\_H\_FRONT is a parameter. The new outputs were added to simplify the animation of payloads and miscellaneous parts attached to the sprung mass. (The origin point of the sprung mass was defined originally as a reference for defining nominal Z locations of wheels supporting the sprung mass.)

Similar front hitch points already existed for the hinged towbar and articulation system types of hitches that were introduced in version 2022.0. Those names were renamed for version 2022.1 to provide consistency, such that the XYZ output variables for all hitch types have the same generic meaning: coordinates of a point fixed in the sprung mass at the height of the front hitch. Table 4 lists the new and old names in the case where the trailing unit is unit 2.

*Table 4. Trailer sprung mass output variables that were renamed (examples for unit 2).*

New Name	Old Name Towbar	Old Name Art System	Description
Xo_FH_2	Xo_Bar_2	Xo_Art_2	X coordinate, front hitch point
Yo_FH_2	Yo_Bar_2	Yo_Art_2	Y coordinate, front hitch point
Zo_FH_2	Zo_Bar_2	Zo_Art_2	Z coordinate, front hitch point

### *Change to name of parameter of electric motor inertia*

The parameter name of the electric motor inertia is changed from IMOTOR to I\_MOTOR in the vehicle model of version 2022.1 to support a new option to include a reduction gear ratio of the electric motor on the **Powertrain: Electric Motor Torque** screen.

### *Updated trailer backing examples*

Past versions included examples of single-trailer vehicles backing up. The controls were customized for each example and applied using VS Commands. Version 2022.1 adds a built-in Trailer Backing Controller that is activated with a single VS Command INSTALL\_DM\_TBC that also adds a single parameter TBC\_GAIN. The new TBC option was used to repeat the older examples.

The following examples have been updated to use the new built-in trailer backing controller:

- Driver Model > Offset Backing: Tractor-Trailer
- Driver Model > Alley Dock
- Driver Model > Reverse DLC: Tractor-Trailer

The examples perform similarly but no longer require the custom steering control VS Commands. These have been re-organized into the new \* **Trailer Backing Controller** category/CPAR. Please refer to the new Help document **Help > Controls > Trailer Backing Controller** for more information.

# TruckSim 2022.0 vs TruckSim 2021.1

## Bug Fixes

The following bugs were identified and fixed in this release.

### *VS Math Model*

1. The speed controller converts a target longitudinal force to a set of target drive torques using estimates of the instant rolling radii of the drive wheels. There are two cases where the wrong wheels were used to obtain the instant tire rolling radii:
  - a. Vehicles with a single drive axle that is the second to last axle on the unit, such as front wheel drive.
  - b. Vehicles with more than five axles used the last axles on the vehicle, instead of the last of the first five, as described in the Powertrain Help document.
1. In both cases, there can be a small effect on the calculated throttle. The powertrain itself works as intended; the effect only effects the controller algorithm.
2. Several changes were made that improve the consistency of how GPS values are calculated for vehicles and moving objects. As a result, GPS outputs in 2022.0 might differ slightly in simulation that cover significant changes in latitude involving multiple vehicles and/or moving objects.
3. The output variable for the compliance contribution to roadwheel steer, `StrC_<wheel>`, was incorrect for the non-steered independent suspension type, always reporting a value of zero.
4. Solid axle and legacy twist beam suspensions had per-wheel outputs for dive, suspension X movement, and suspension lateral movement (e.g., `Dive_L1`, `SusX_L1`, `SusY_L1`). These outputs are used by other suspension types but do not make sense for the solid axle or legacy twist, which have per-axle definitions. Moreover, these outputs were always zero for these cases. For solid axles, these outputs have been replaced with per-axle values (e.g., `Dive_A2`, `SusX_A2`, `SusY_A2`) which are consistent with the solid axle kinematics definitions. For legacy twist beam suspensions, the per-wheel outputs are removed.
5. For independent suspensions, the output variable `CamUS<wheel>`, which gives the camber angle of the unsteered unsprung mass (i.e., the kingpin inclination change), was calculated incorrectly.
6. For independent suspensions, there has been an improvement in the formulation of the multibody constraint involving the camber as a function of jounce, configurable function root keyword `CAMBER`. The camber angle values from this input table are applied as before, but the equations involving the partial derivatives of this table have been updated for improved accuracy and efficiency. This will produce slight differences in simulation results for runs using a vehicle with independent suspension.

7. For independent suspensions, the output variable `Dive_<wheel>`, which gives the dive angle of the unsteered unsprung mass (i.e., the pitch angle generating caster change), was calculated slightly incorrectly.
8. For independent suspensions, there has been an improvement in the formulation of the multibody constraint involving the dive as a function of jounce, configurable function root keyword `SUSP_DIVE`. The dive angle values from this input table are applied as before, but the equations involving the partial derivatives of this table have been updated for improved accuracy. This will produce slight differences in simulation results for runs using a vehicle with independent suspension.
9. For calculations using a clothoid, including segments in the Path: Segment Builder, there has been an improvement for calculations along the clothoid when the entering and exiting curvature are non-zero values.

### **VS Visualizer**

1. Numerous plotting fixes and improvements, including:
2. Fixed problems with plotting continuous expressions (such as `"sin(Time)"`).
3. Improved plotting of first and 2nd derivative values of Data Channels (e.g. `"Xo.dydx"`, `"Xo.d2ydx2"`).
4. Improvements and fixes to reloading data for plots containing algebraic expressions.
5. Fixed bug where plots created using drag-and-drop did not have a label in the plot legend.

### **VS Scene Builder**

The 64-bit version of the VS Scene Builder was unable to open VS Scene Builder files generated with the 32-bit version of the VS Scene Builder and vice versa. This was fixed; each version can now read files generated by the other.

## **Backward Compatibility**

### ***Change to name of `mfsswift_carsim_wrapper.dll` (MF-Tyre and MF-Swift)***

For the TruckSim 2021.1 release, the VS STI module used to support Siemens tire models (MF-Tyre and MF-Swift) included the Mechanical Simulation product release number: `mfsswift_carsim_wrapper_2021_1.dll`. For the 2022.0 release, the version number has been removed. End-users who open a TruckSim 2021.1 database with 2022.0 will have to browse to the **Tire (External)** screen, select their MF-Tyre and/or MF-Swift datasets, and update the contents of the VS STI module (DLL) field to use the new wrapper name: `mfsswift_carsim_wrapper.dll`. On the **Tire (External)** screen, the updated file path will be: `Extensions\External_Tire_API\mfsswift_carsim_wrapper.dll`.

For reference, the `.dll` is in the `TruckSim_Prog` folder:

```
TruckSim2022_Prog\Resources\Extensions\External_Tire_API\  
mfsswift_carsim_wrapper.dll
```

### *Changes in names of state variables for trailer sprung masses*

A few state variables for trailer sprung mass motions were renamed to provide better consistency. Table 5 lists the names that apply for the first trailer: unit 2. Similar names are used for other trailers. These changes are mainly of interest for those who might use the variable names in VS Command equations.

### *Hitch variables only defined when used*

Calculations made for hitch connections were redone to provide better support for hitch types other than the fifth wheels commonly used with heavy vehicle combinations. A new command `OPT_HITCH_TYPE` sets the type of hitch to 1 (generic or fifth wheel), 2 (ball or pintle hitch), 3 (ball/pintle connected to a massless tow bar, where the tow bar connects to the trailing vehicle unit's sprung mass body, i.e., a dolly), or 4 (articulation system, with a pitch hinge located in front of an articulation pivot).

Some of the hitch-related output variables are only applicable for certain types of hitches. Variables for the first type (fifth wheel and generic) were always created in past versions; starting in 2022.0, they are only created if `OPT_HITCH_TYPE` is set to 1 or 4.

*Table 5. Trailer sprung mass state variables that were renamed or replaced.*

New Name	Old Name	Description
SV_AVX_2	SV_AVX_SM2	Body-X roll rate of sprung-mass 2 {AVx_2}
SV_AY_2	SV_AY_SM2	Body-Y pitch rate of sprung-mass 2 {AVy_2}
SV_AVZ_2	SV_AVZ_SM2	Body-Z yaw rate of sprung-mass 2 {AVz_2}
SV_PITCH_REL_2	SV_PITCH_H1	Relative pitch (3-2-1) for unit 2
SV_ROLL_REL_2	SV_ROLL_H1	Relative roll (3-2-1) for unit 2
SV_YAW_REL_2	SV_YAW_H1	Relative yaw (3-2-1) for unit 2
SV_VXS_2	SV_VX_SM2	Body X speed of laden sprung-mass CG_2
SV_VYS_2	SV_VY_SM2	Body Y speed of laden sprung-mass CG_2
SV_VZS_2	SV_VZ_SM2	Body Z speed of laden sprung-mass CG_2

Table 6 lists output variables that are only created for a hitch when (`OPT_HITCH_TYPE` = 1).

Table 6. Output variables when *OPT\_HITCH\_TYPE* = 1.

Short Name	Long Name	Units	Type
Mx_Hh	X 2-1-3 moment, hitch <i>h</i>	N-m	Hitch Moments
My_Hh	Y 2-1-3 moment, hitch <i>h</i>		
Mz_Hh	Z 2-1-3 moment, hitch <i>h</i>		
Pitch_Hh	Pitch, 2-1-3, hitch <i>h</i>	deg	2-1-3 Euler Angles
Roll_Hh	Roll, 2-1-3, hitch <i>h</i>		
Art_Hh	Articulation, 2-1-3, hitch <i>h</i>		
PitchR_Hh	Pitch rate, hitch <i>h</i>	deg/s	2-1-3 Euler Angle Rates
RollR_Hh	Roll rate, hitch <i>h</i>		
ArtR_Hh	Articulation rate, hitch <i>h</i>		

State variables have been used to support torsional friction in hitches. These require the moment at the last time step and the angles at the previous time step. Table 7 lists the state variables that were always included in past versions for hitch 1 (i.e., the connection between the lead unit and the first trailing unit) but are only included in new versions when *OPT\_HITCH\_TYPE* is set to 1 or 4.

Table 7. Hitch state variables that are created only for hitches with torsional moments.

Name	Description
SV_MX_H1	Roll moment minus damping, hitch 1
SV_MY_H1	Pitch moment minus damping, hitch 1
SV_MZ_H1	Articulation moment minus damping, hitch 1
SV_PITCH_H1_OLD	Relative pitch (2-1-3) for hitch 1
SV_ROLL_H1_OLD	Relative roll (2-1-3) for hitch 1
SV_ART_H1_OLD	Articulation (2-1-3) for hitch 1

The same conditions apply for three import variables that provide hitch moments from external models. Table 8 lists import variables that were always available in past versions for hitch 1 but are only included in new versions if they can be applied.

Table 8. Hitch import variables that are created only for hitches with torsional moments.

Name	Description
IMP_MHX1	Roll moment, 2-1-3, hitch 1
IMP_MHY1	Pitch moment, 2-1-3, hitch 1
IMP_MHZ1	Articulation moment, 2-1-3, hitch 1

## Known Issues

### *OpenDrive scene tiles built with VS Scene Builder do not display in VS Visualizer*

The 2022.0 version of VS Scene Builder does not generate the required road FBX file in the correct directory when OpenDRIVE files are imported by the user. The result is that the road is not displayed in VS Visualizer. This has been fixed for VS Scene Builder 2022.1. The workaround is to set the OpenDRIVE Tile Import Location on the Options screen to any directory that is immediately under a \Resources directory (for example: C:\Users\User name\Resources\OpenDrive).

## TruckSim 2021.1 vs TruckSim 2021.0

### Bug Fixes

The following bugs were identified and fixed in this release.

#### VS Browser

1. An intermittent crash was identified and fixed involving “mix tables” whose cells are not all simple fields; they support GUI elements such as ring controls and blue links (e.g., **Path: Segment Builder** screen). The problem occurred when clicking Blue Links within a mix table. This bug existed for versions 2020.1 to 2021.0.
2. An intermittent crash was identified and fixed involving the pasting of large text strings into Yellow Fields.
3. Several of the GUI screens had overlapping text with the tables.
4. The License Source Selection dialog could pop back up after clicking Cancel.
5. In the 2021.0 release, the **File** menu option **Export Encrypted Expanded Parsfile** created the expanded parsfile without encryption. The **File** menu option **Import Parsfile (Any Export Type)** handles the imported expanded Parsfile as an ordinary (not encrypted) expanded Parsfile.

**Warning** There is no workaround for this bug. If encrypted data is needed, do not use version 2021.0.

6. The **Animator Shape File Link** and **Animator Shape Assembly** screen have rows for scale factors and angle offsets applied to the shape(s) for the dataset. The order shown on the screen was changed to match the order in which the transforms were applied: scale first, then rotation. The change is only in appearance; existing datasets are not affected.
7. Fixed a crash when pasting large blocks of text into yellow fields.
8. Fixed a collection of issues involving missing or misaligned UI elements when editing tables.
9. Removed unnecessary popup messages when a license is missing or expired.

## VS Math Model

10. Many problems involving invalid VS Commands have been fixed, such that the VS Math Model will now generate a descriptive error message and the math model will not crash.
11. If the Speed Controller was not installed, the Echo file showed all 200 SPEED\_TARGET Configurable Function datasets. The visibility of each SPEED\_TARGET dataset is now properly set based on the value of the parameter N\_SPEED\_TARGET, whether or not the Speed Controller is installed.
12. The steering system options OPT\_RACK, OPT\_M\_SW\_CALC, OPT\_STEER\_DEF, and OPT\_I\_GEAR\_IN formerly allowed settings other than 0 or 1. There is now an error message should these options be given invalid settings.
13. The output variable Ax\_SCcmd (the acceleration command in the speed controller) is now set to 0 when acceleration control is disabled (when OPT\_SC  $\neq$  5).
14. When using the VS API function vs\_run to repeatedly make a run that includes the path detector, the run could fail with an error message referring to installing the path detector more than once. This is fixed.
15. If the parameter LTARG\_ID\_DM was changed in an Event, it did not force a reset for the driver model or speed controller in path-preview mode (OPT\_SC = 4).
16. The parameter Y\_CG\_SUSP was incorrectly shown in the Echo file for non-solid-axle suspension types. This now appears only for axles using the solid axle model type.
17. Several optional solid axle parameters did not support indexing with the IAXLE or ISIDE keywords: SPG\_AX, SPG\_AY, and SPG\_LEN. These keywords now support this form of indexing.
18. Math model parameters which are read-only are now flagged if they are used in a VS Define Command, e.g. DEFINE\_PARAMETER, DEFINE\_OUTPUT, etc. These include the System Constants DR (degree/radian), G (gravity), PI, and ZERO.
19. With the hybrid powertrain in prior versions, the engine is always operated based on the internal condition although the hybrid mode (IMP\_HYBRID\_MODE) is imported as replace (external command is ignored). And this is fixed.
20. If an encrypted parsfile is used for a run, the Math Model will probably generate an error message about an invalid input. This behavior was caused by a combination of new parsing logic in 2021.0 and a problem with generating encrypted Parsfiles (see the previous subsection), such that decryption was not tested as intended at Mechanical Simulation.

As mentioned earlier, if encrypted data is needed, do not use version 2021.0.

## Documentation

The **Help > Driver Models** document had an incorrect equation for the acceleration mode of the speed controller. The incorrect equation was:  $A_{xscrq} = K_{acc} * (A_{xscrq} - A_{xrd})$ . The correct equation, which matches the internal VS Math Model code, is:  $A_{xscrq} = A_{xrd} + K_{Acc} (A_{xSCcmd} - A_{xRd})$ .



## Backward Compatibility

### *Changes to S-L from X-Y iterative solver settings*

The new default settings for calculating S-L coordinates from X-Y values via iteration will change simulation results, given the slight differences in solution for station compared to the old settings. The default settings as of 2021.1 are OPT\_SL\_METHOD=1 and TOL\_SL\_METHOD=1 mm. To reproduce the old solutions for station (pre-2021.1 behavior), use a miscellaneous yellow field to change these settings to OPT\_SL\_METHOD=0 and TOL\_SL\_METHOD=5 mm.

### *Changes to default steering system properties*

The default values for the following steering system properties have been updated: Column inertia (VS Math Model keyword I\_COL), system inertia (I\_GEAR\_IN), column damping (D\_COL), column hysteresis (HYS\_COL), and hysteresis reference angle (BETA\_COL). The new default values are similar in magnitude to the example vehicles included in your VehicleSim product. If you have a vehicle which uses default values, and you import it into version 2021.1, you may observe differences in simulation results. To restore the previous behavior, you may enter the values shown in Table 9 on an appropriate steering screen or in a miscellaneous yellow field.

*Table 9. Changed default values for steering parameters.*

Parameter	Default value before 2021.1	Default value as of 2021.1
I_COL	0.000001	0.01
I_GEAR_IN	0.000001	0.0001
D_COL	0.0	0.01
HYS_COL	0.0	0.1
BETA_COL	2.0	0.1

### *Effects of solid axle improvements*

Version 2021.1 introduces new features to the solid axle suspension model — new 2D table types, nonzero axle pitch inertia, nonzero axle X CG offset — which necessarily involve changes to the internal calculations of the VS Math Model. These new features are optional and there is no conceptual change in behavior when they are not used; however, you may notice some extremely small differences, nearing machine precision, in simulation results.

Moreover, solid axle suspension examples have been updated to use the new library screens of **Suspension: Dive Angle (Solid Axle)** and **Suspension: Longitudinal Movement (Solid Axle)**. Existing solid axle datasets will import with no issues, but the blue links for library screens which are no longer recommended for solid axles will now show as orange on the **Suspension: Solid Axle System Kinematics** screen.

### *Discontinuation of TNO Tire Models*

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. Datasets of **Tire (External)** library with those abolished model options from prior versions can be imported

and automatically assigned to either **Siemens MF-Tyre Only** or **Siemens MF-Tyre/MF-Swift** model option.

Siemens MF-Tyre/MF-Swift model options were not available in prior versions of TruckSim however they are available only for Windows from this release.

## TruckSim 2021.0 vs TruckSim 2020.1

### Bug Fixes

The following bugs were identified and fixed in this release.

#### *VS Math Model*

1. The import variable `Imp_Throttle_Engine` was ignored when the automatic clutch control was used via the screen **Control: Clutch Shifting Timelines (Closed Loop)**.
2. The built-in electronic stability controller (ESC) had a bug which caused the brake pressure values calculated for the rear wheels to be applied to the wrong side of the vehicle (for example, the left rear brake signal from the ESC was sent to the right rear brake).
3. For vehicles with trailers, setting `OPT_INIT_SPEED = 1` when `OPT_INIT_CONFIG = 0` caused the solver to crash. These settings are intended to be independent from each other. The bug began in version 2018.1 and is corrected for version 2021.0.
4. Changing a vehicle's `OPT_PT` setting after it had already been defined caused an error message mentioning the keyword `OPT_MOTOR_ON_AXLE` followed by a crash. In this circumstance, the error message now gives a more accurate description, and the software no longer crashes.
5. The calculation which sets the static wheel alignment from keywords `A_TOE` and `A_CAMBER` had a small missing term, leading to minute differences (~0.01%) in static camber and toe when comparing what was entered on the suspension kinematics screen versus what was output at the first timestep (prior to the springs settling).
6. User-defined functions are usually applied in symbolic expressions in equation lists (`EQ_IN`, `EQ_DYN`, etc.) or for assigning values to parameters. If applied as a command statement, e.g., `MY_FUNC (1, t)`, they are supposed to be applied, even though a returned value is not saved. This worked if the command was put into an equation list, but not if it was simply typed into a Parsfile. For example, `EQ_INIT MY_FUNC (1, t)` worked, but typing `MY_FUNC (1, t)` in a yellow miscellaneous field did not.
7. When loading a 2-D table from a VSTB file with the command `LOAD_TABLE_FILE`, the units for the X and Y axes were not properly scaled if the units had an associated gain (e.g., degrees).
8. Path curvature obtained with VS Commands functions like `PATH_CURV_ID` or provided by the output `Rho_Road` have a discrepancy for looped paths defined using X-Y table data. They sometimes show different curvature values at the first and last points of the path, despite the assumption that these represent a single point on the loop. This does not affect curvature that is calculated using the path preview speed controller or path detector.

9. The import names for aerodynamic forces and moments have a suffix identifying the sprung mass if the model has more than one unit, e.g., for unit 2 the imported longitudinal force is `IMP_FX_AERO_S2`. For unit 1, there should be no suffix. However, there was a suffix `_S1` if there were trailers or other vehicles. The names for unit 1 have been fixed so there is never a suffix.
10. If the simulation had more than one vehicle and there is a multiple vehicle license, it should support four vehicles and almost any number of trailers. However, an error message was generated when adding a vehicle if there were already four units, including trailers. (The check is not supposed to count trailers.)
11. In version 2020.1 only, twin clutch differential state variables and twin clutch differential output variables (e.g. `ClutchL3`, `ClutchR3`) are written for all differentials and transfer cases, but twin clutch differentials are only supported on axles 1 and 2. For transfer cases, these state variables and outputs have malformed descriptions. The variables have been removed for 2021.0, except for differentials that support the twin clutch feature.
12. For 6WD, 8WD, and 10WD powertrains, malformed variables were removed (Table 10. Malformed variables that were removed).

*Table 10. Malformed variables that were removed from 6WD, 8WD, and 10WD powertrains.*

Import Variables	Output Variables
<code>IMP_AV_</code>	<code>AV_</code>
<code>IMP_M_OUT_</code>	<code>M_</code>
<code>IMP_ROT</code>	<code>Rot_</code>

13. The slope of the road had incorrect sign when using VS Terrain under certain conditions. The sign of the pitch (`PitchG*`) and roll (`RollG*`) of the ground was wrong when the tire was using a single point contact patch. The sign was also incorrect for the road aero reference point slopes `dZdXair` and `dZdYair`.
14. An internal calculation for wheel speed on vehicle 2+ (multiple vehicles per solver) was incorrect and could affect powertrain shifting behavior for 2<sup>nd</sup> and higher vehicles. The wheel speed calculation has been corrected for 2021.0 to work with any number of vehicles in the simulation.
15. In version 2020.1, the closed loop steer by torque option for the driver model did not function correctly when `OPT_DRIVER_ACTION = 1` (the default setting). Additionally, since version 2019.1, the behavior of closed loop steer by torque when `OPT_DRIVER_ACTION = 0` has been undefined, despite an attempted fix in version 2020.1. In version 2021.0, when in closed loop torque control mode with `OPT_DRIVER_ACTION = 0`, the angle from the driver, `Steer_DM`, is calculated but unused, and users are expected to import steering wheel torque. See the Driver Control Screens document for more details.
16. The internal calculation of the kinematic steering ratio had a check for dividing by zero which did not function correctly. This has been updated. Typical steering system

kinematics data should not produce the divide by zero condition, so this will not affect typical simulation results.

17. The import variable for the transmission output speed, `IMP_AV_TRANS`, was disabled for the hybrid and electric powertrain options in the previous versions and this is enabled in version 2021.0.

### *VS Browser*

1. The behavior of static reference length scaling for animator shapes did not match what was described in the documentation. The new behavior is that animator shapes will only be Statically Scaled if both a Length (e.g., `X_Length`) and a Reference Length (e.g., `X_Ref_Length`) are specified. Although most of the shipping examples are unaffected, some of the reference length data had to be updated on **Animator: Shape File Link** screens and **Animator: Reference Frame** screens. If databases are converted from 2020.1 or older and animation shapes appear to have incorrect scaling, check the corresponding example in the 2021.0 shipping databases to see what changes have to be made to the imported datasets.
2. For the 2020.1 release two new segment types were added to the **Path: Segment Builder** screen: `close gap (cubic)` and `close gap (inline)`. These were intended to be used as the final segment of a path as a means of closing any gaps that remain when using any of the other segment types. If either of these were selected as the first segment type, the VS Browser would crash. For 2021.0, an error message is prompted if either of these are selected as the first segment.
3. The **Steering Column and Assist** library screen would not write the torsion bar stiffness value if the power steering system was selected as column assist; the default value of 2 N-m/deg was always used. This is fixed.

### *Documentation*

The new features document for version 2020.1 erroneously described the electronic stability controller (ESC) as being extended for use with multiple vehicles running within a single simulation. The capability to support multiple vehicles was introduced in 2021.0.

## **Backward Compatibility**

### *Improvements to Error Handling*

Error handling has been improved such that VS Commands with typos will now generate an error message showing the offending formula (e.g., a space in a variable name). In earlier versions some errors were not reported, resulting in simulations that may not have processed the VS Commands as expected.

The syntax is now checked more thoroughly; errors are reported, and the simulation will stop. In addition to popup messages when running on Windows, the error is reported in the Log file generated with each simulation. If a formula generates a numerical error (divide by zero, use of an undefined variable, etc.), the error message will show the formula.

### *Indexing of Powertrain Math Model Keywords*

Starting in 2021.0, the indexing of some powertrain math model keywords have had their indices updated to reflect the potential installation of one or two electric motors per axle, e.g., for a single motor RWD vehicle, `MMOTOR_MAX_GAIN(1)` is now `MMOTOR_MAX_GAIN(1,1)`. Under most conditions this has no effect. However, if any of these keywords were used in VS Commands in 2020.1, those VS Commands will not work when the example is used in 2021.0, because the index will be incorrect. To regain the intended functionality, the affected keywords will have to be provided a proper index value in the VS Commands. These can be determined by searching the Echo file for the math model keyword and noting what indices are assigned to them.

### *Kingpin Rotation Angle Calculation*

For suspensions which have steer angles defined in vehicle coordinates, the VS Math Model converts the steer kinematics from the suspension/steering tables into an equivalent rotation about the inclined kingpin axis. This kingpin rotation angle is then used as a constraint in the multibody suspension model.

This calculation was replaced beginning in version 2021.0 with a more efficient version that does not rely on iteration. Consequently, the model will no longer fail during the run with the error message, “...there was a problem in *kp\_constraint*: could not calculate the kingpin steer angle...”. Error handling for the new calculation method occurs at initialization, by checking the total inclination of the kingpin axis with respect to vertical and making sure it does not exceed 45 degrees.

The 2021.0 calculation method produces slight differences in simulation results, but these are attributable to the accuracy of the new calculation compared to the previous iterative calculation, which would produce results only as accurate as its solution tolerance allowed.

### *Handling Course Racing Line*

Runs involving the Handling Course with a separate X-Y path for the racing line have been updated with new initial and final X-Y coordinates, thus aligning the racing line with the road's initial and final X-Y coordinates. In versions 2020.1 and older, the racing line's initial / final X-Y coordinates were about 5 meters in front of that of the road. This did not negatively affect the simulation but could invite questions as to why the vehicles were starting and stopping in front of the road's finish line.

### *Variable Name Changes for 6WD Powertrains*

Table 11 lists variable names that were changed for 6WD powertrains. For 6WD, the first differential connects directly to the 7th differential (master transfer case), not the third differential.

Table 11. Variables whose names were changed for 6WD powertrains.

Old Import Variable	New Import Variable
IMP_AV_D3_F	IMP_AV_D7_F
IMP_M_OUT_D3_F	IMP_M_OUT_D7_F
IMP_ROT_D3_F	IMP_ROT_D7_F
Old Output Variable	New Output Variable
AV_D3F	AV_D7F
M_D3F	M_D7F
Rot_D3f	Rot_D7f

### *Updated Road Wheel Steer Kinematics for SS\_SS Vehicles*

Based on the work done for the new 8x8 LCF Van with the 16-speed manual transmission, all 4-axle SS\_SS vehicle examples in TruckSim use the same axle positions and road wheel steer kinematics data for Axles 1 and 2. During the development of the new vehicle, it was found that the previously used road wheel steer kinematics data was acceptable for small steering wheel angles (i.e., a high speed double lane change) but performed poorly with large inputs, such as what might be encountered during low speed maneuvers (i.e., negotiating an intersection or a slowly increasing steer test). Since the data was not matched well to the wheelbase of the truck, it was found that Axle 2 steered more than Axle 1; this resulted in counteracting tire lateral forces that diminished the vehicle's ability to negotiate small radius turns and led to unreasonable steering torque feedback. The new road wheel steer kinematics data works to address these deficiencies.

For users upgrading from old databases, it is recommended to use the new road wheel steer kinematics data and axle positions. A new steering torque dataset (from the Library **Steering > More > Steering Wheel Torque**) is matched to the new steering system data. If your vehicle's axle positions and/or track width are different than the shipping examples, the road wheel steer kinematics data should be recalculated to match the data used.

### *3-Axle Dump Trucks with No Brakes Linked to Axle 2*

TruckSim has included several examples of a 3-Axle Dump Truck for many years, with variations including a payload, locked Transfer Case, and towing a dolly / trailer combination. Two of the dump trucks inadvertently had the brake system for Axle 2 set to "No brakes." Although this had no effect on the Run Control examples in which they were used, the lack of a brake system on Axle 2 would produce longer stopping distances than an equivalent vehicle with brakes linked. For TruckSim version 2021.0, these trucks were replaced with vehicles that include a brake system linked to Axle 2. As part of this effort, several duplicate datasets were removed while others were renamed.

### *Vertical Offsets for Commercial Vehicle Animator Assets*

Many of the commercial vehicle animator assets were updated for the 2021.0 release. This effort included eliminating the need for the vertical offsets that were typically applied on the **Animator: Shape File Link** screen. For animator assets that have a visible vertical offset following a database

conversion from versions older than 2021.0, please refer to the shipping database included with the 2021.0 installation to see what changes need to be made to the **Animator: Shape File Link** datasets.

## Known Issues

### *Export Encrypted Parse File Feature is Nonfunctional*

The 2021.0 version of the VS Browser shipped with a bug where the encryption step exports a plain unencrypted \*.par file. There is no current workaround for this issue. Customers relying on the encryption feature should use an older version (2020.1 or earlier) or upgrade to a later version (2021.1 and later) with the code fix.

## TruckSim 2020.1 vs TruckSim 2020.0

### Bug Fixes

The following bugs were identified and fixed in this release.

#### *VS Math Model*

3. The parameter `N_LTARG` specifies how many LTARG datasets are visible in the Echo file. This number is initially zero, and it is typically incremented as LTARG datasets are added using the VS Browser which writes the VS Command `SET_ILTARG_FOR_ID` in the dataset Parsfile. However, if a run is continued using the `End.Par` file, and the text from the Echo file is `N_LTARG 0`, an error window popped up declaring that the value must be 1 or greater. The lower limit is 0, not 1.
4. The parameter `N_MIN_DETECTIONS` specifies a minimum number of moving objects that will eventually be defined. It is used when defining a new ADAS Sensor, such that detection variables will be defined immediately for use in VS Commands, possibly before the Objects have been defined. In past versions, this parameter did not exist until the first Sensor was defined. Therefore, it was only applied when other sensors were defined. Attempts to set the value of `N_MIN_DETECTIONS` before the first sensor was defined were ignored.
5. Several functions available for use in VS Commands that involve road properties did not work as expected when VS Terrain was used to specify ground geometry. Functions such as `ROAD_Z_ID` and `ROAD_DZDS_ID` returned zero. The functions listed in Table 8 in the *Paths and Road Surfaces Help* document now function when VS Terrain is in use with the substitution that `S = X` and `L = Y`. In a related change, the output variable `Station` is now automatically set to `X` when using VS Terrain.
6. Deprecated functions for use in VS Commands listed in Tables 10 and 11 in the *Paths and Road Surfaces Help* document generally do not work when VS Terrain is in use. VS Terrain did not exist when they were deprecated, so no changes will be made.
7. The speed controller has an acceleration command option, identified with the parameter `OPT_SC = 5`. In this mode, speed-related parameters are hidden, as is the option to use the

- brake system as specified by `OPT_BK_SC`. The intent is that the brakes will always be used in this mode when the acceleration command is negative. However, there was a bug in the code such that the brakes would only be applied for negative command acceleration if the hidden parameter `OPT_BK_SC = 1`.
8. The speed controller automatically turned off integral control with driving in reverse. That has been changed such that integral control now works in reverse. The controller now allows the vehicle to reach constant target speeds in both forward and reverse directions.
  9. If external springs were specified for a suspension and the auxiliary roll stiffness was zero, the simulation would crash during initialization when trying to calculate static loads.
  10. The Boolean operators ‘&’ (AND) and ‘|’ (OR) that can be used in symbolic formulas for VS Commands did not always return 0 or 1 when applied. The AND operator returned either 0 (correctly) or the value of the second argument (should be 1 if the argument is nonzero). The OR operator returned 0 (correctly), returned 1 if the first value was nonzero (correct), or the value of the second argument (should be 1 if the first argument is 0 and the second is nonzero). They now provide only 0 or 1 results, as documented.
  11. External Tire Models (MF-Tyre/MF-Swift, COSIN FTire, and Michelin TameTire) could not be used on trailers with two or more axles. For version 2020.1, support for External Tire Models on trailers has been extended for up to seven vehicle units and 28 tires. This issue had existed since version 2019.0.
  12. The `TLAG_CL_TWIN` parameter (twin clutch time delay) was ineffective and behaved as a zero delay due to a bug in versions 2019.0 through 2020.0.
  13. During powertrain initialization, such as at the beginning of a run or when triggered by an Event, the import variables `IMP_MODE_TRANS` and `IMP_GEAR_TRANS` were mistakenly applied twice. For example, if one of these was used in the “Add” mode, the imported value would get added to the native value two times during the initialization. This behavior was not present outside of the initialization phase.
  14. If the import variable `Imp_Mode_Trans` was set to -1 with the intent of driving the vehicle in reverse and the Single Point Driver Model was used, the powertrain would receive the correct information but the Driver Model would not. The result was that the Driver Model Preview Point initialized in the incorrect location.
  15. When obtaining ground contact information ( $Z$ ,  $dZ/dX$ ,  $dZ/dY$ ) using import variables, the  $dZ/dX$  and  $dZ/dY$  components could have errors if the internal tire model was set to use non-zero values for the contact patch length and width.
  16. The parameter `OPT_DRIVER_ACTION` was ignored when the Closed Loop Driver Model was in torque control mode.
  17. When using VS functions to interact with reference paths or roads, a query ID (*qid*) should be used to track a single point of interest moving along one or more paths. It was intended that the *qid* can be set to a number from 1-99. However, for TruckSim versions 2020.0 and older, setting the *qid* to 1 could cause the simulation to fail. (The *qid* could be set to a number from 2-99.)



**Note** The query ID issue was not mentioned in the 2020.1 version of this document; it was added June 2021.

18. In version 2020.0, the VS Browser could crash during the import of an encrypted expanded parsfile. This has been fixed for version 2020.1.

### *VS Browser*

1. Importing an Expanded Parsfile from TruckSim 8 or older caused a crash in 2020.0.
2. When the Configurable Function option “Spline interpolation & extrapolation” was selected, the plot shown in the GUI would be improperly displayed if values crossed the X-axis.
3. The **Tire** screen did not support 64-bit versions of MF-Tyre/COSIN model. The screen now supports 32 and 64-bit versions of MF-Tyre/COSIN model by loading the appropriate wrapper .dll (either 32 or 64 bit) regarding the bit type of what is specified in the field for the tire solver (.dll) path.

### *VS SceneBuilder*

VS SceneBuilder 2020.0 road lane markings were not generated in either the tile thumbnail or the FBX shape file.

## **Backward Compatibility**

### *VS Terrain*

When VS Terrain is used in a Solver, the older VS Road Surface model is turned off. In past versions, API functions and VS Command functions that involved the road reference path used by the current road did not have consistent behavior. In 2020.1, a road path is assumed when VS Terrain is in use. The VS Terrain path is straight and oriented such that S = global X and L = global Y.

### *Inclined Dampers for Solid Axles*

Solid axle suspensions have supported inclined dampers since versions 8.1. In order to have the version 8.1 damper model approximate the pre-8.1 behavior, a default damper length of 10 m was established for solid axle dampers. Beginning in TruckSim 2020.1, the pre-8.1 behavior is available by setting damper length parameter DMP\_LEN equal to zero. Specifying a length greater than zero uses the inclined damper behavior introduced in version 8.1. Because none of the example vehicles provided with TruckSim used a nondefault damper length, the default value has now been set to zero to enable the pre-8.1 behavior. The damper calculation method used from versions 8.1 to 2020.0 is available by manually setting each DMP\_LEN parameter equal to 10000 mm (10 m).

### *Pacejka 5.2 Magic Formula Tire Model*

Support for the Pacejka 5.2 Magic Formula tire model is deprecated, and this feature is now supported by the Siemens/TASS external tire model. When the PAC52 tire model is selected on

the **Tire** screen, a button is available to export a .tir format file from the linked parameters. Instructions for connecting to the external tire model are available from **Help>Deprecated Screens>Internal Pacejka 5.2 Tire**.

### *Road Coefficient of Friction Dataset for K&C Tests*

For 2020.1, all K&C tests, 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 databases. These may be removed in a future release.

### *Indexing of Powertrain Math Model Keywords*

Starting in 2020.1, some math model keywords associated with the powertrain are now indexed, e.g., `MMOTOR_MAX_GAIN(1)`. Under most conditions this change has no effect. However, if any of these parameters were used in VS Commands in 2020.0, those VS Commands will not have the same effect after a database has been upgraded to 2020.1. To regain the intended functionality, the affected parameters will have to be provided with the proper index value. These can be determined by searching an Echo file for the math model keyword and noting what indices are assigned to them.

### *Vehicle Lead Unit screens*

The **Vehicle Lead Unit** screens used to represent a lead vehicle unit now increment an indexed parameter `IVEHICLE` in support of simulations with multiple vehicles. In past versions, any link to a dataset from a **Vehicle Lead Unit** library after the first link (from the **Run Control** screen) would override parameters and tables related to the lead unit. Starting with 2020.1, each link to a **Vehicle Lead Unit** dataset adds a vehicle to the simulation.

All shipping databases in past versions would have a single link to a **Vehicle Lead Unit** dataset for each run, so no compatibility issues are expected. However, if there are custom user-defined setups with multiple links to **Vehicle Lead Unit** datasets, then the newer versions will attempt to activate multiple vehicles.

### *Links to Moving Object datasets*

In past versions, Moving Object datasets that link to a vehicle animation shape should not have been linked to datasets that were scanned before the vehicle datasets. Doing so caused animations to be incorrect.

Links that could attach to Moving Objects have been revised in some cases such that those links are no longer valid. Existing datasets updated to 2020.1 will show red links, indicating that a problem existed and should be fixed.

### *Rolling resistance coefficient RR\_SURF*

In past versions, a road parameter RR\_SURF defined the surface coefficient for tire rolling resistance and was fixed for a given road surface. The parameter has been replaced with a 2D table that can provide RR\_SURF as a function of road S and L coordinates. If VS Terrain is used, RR\_SURF is provided as a function of X and Y coordinates.

If RR\_SURF is specified using the yellow data field on the **Road: 3D Surface (All Properties)** screen, existing datasets will work as intended. However, if RR\_SURF is specified in a Miscellaneous yellow field, the reference should be replaced with a dataset from one of the two RR\_SURF libraries: **Road: Rolling Resistance Map, S-L Grid** or **Road: Rolling Resistance, Variable Width**.

### *Control: Shifting (Closed Loop) screen*

As of version 2019.1, the screen **Control: Shifting (Closed Loop)** was updated such that it was no longer possible to link directly to the **Control: Shifting (Open Loop)** screen using the drop-down menu in the lower left corner. However, if a database or .cpar was created in version 2019.0 or earlier that included a link to the **Control: Shifting (Open Loop)** screen from the **Control: Shifting (Closed Loop)** screen and then it is opened with version 2020.1, the **Control: Shifting (Open Loop)** dataset will still be in the screen parsfile for the **Control: Shifting (Closed Loop)** screen. The effect is that the data specified on the **Control: Shifting (Open Loop)** screen will still be used in the simulation.

For version 2020.1, the recommended practice is to break the link to the **Control: Shifting (Open Loop)** dataset by toggling the library link for the Closed-loop shift control settings (i.e., select [No linked library]). If the simulation is supposed to use a dataset from the **Control: Shifting (Open Loop)** library, then that link should be made on the **Procedure** or **Event** screen.

<b>Note</b>	This issue originally occurred as of the 2020.0 release, and that section of this document has been updated as of the 2020.1 release. This issue is mentioned again here because for the 2020.1 release, the link to the <b>Control: Shifting (Open Loop)</b> screen will be flagged as a broken link.
-------------	--

### *Tire/Ground Friction Coefficient (Mu) Outputs from External Tire Models*

When external tire models are used, the existing tire/ground friction coefficient (Mu) outputs, MuX\_ and MuY\_, used to be overwritten by the values from the external models. Starting with 2020.1, the following changes have been made:

1. New outputs, ExMuX\_ and ExMuY\_, which are defined by the external tire model are added only when the external tire model is in use, and
2. Existing tire/ground friction outputs, MuX\_ and MuY\_, show the friction coefficient from the VehicleSim road.

### *The VEHICLE\_CODE in a Simfile*

In past versions, the vehicle model was determined by information in the Simfile associated with the keyword `VEHICLE_CODE`. In 2020.1, that information is provided by information in the Parsfile associated with the keyword `MODEL_LAYOUT`.

If running from the TruckSim browser, this change is handled behind the scenes and all existing runs should have the same layouts as in past versions. However, if the VS Solver is loaded and run from external software, the preparation of the Simfile and All Parsfile will probably need to be changed.

### *Steering Controller with External C and Simulink*

Changes to the VS Solver for 2020.1 mean that this example from a 2020.0 or older database will not work once the database is converted to 2020.1. For end-users who are converting an older database and want to use this example in 2020.1, the recommendations are:

1. Use the example that is included with the 2020.1 database, or
2. Follow the instructions in the `Readme.txt` file to rebuild the S-Function wrappers (TruckSim\_Data\Extensions\Custom\_C\S-Function).

### *Orientation of Animation Shapes*

The shape rotation and offset for the overhead traffic signals and their associated components was changed between 2020.0 and 2020.1 such that +X emerges from the front face of the sign or signal, +Y emerges from the left side, and +Z is up.

For end-users upgrading an older database, the options for fixing the animation shapes are:

1. Compare the **Animator: Shape File Link** datasets between the upgraded database and a clean 2020.1 database, and make the adjustments manually, or
2. Replace the **Animator: Shape File Link** datasets in the upgraded database with those from a clean 2020.1 database.

### *Powertrain Variables*

Table 12 lists variables for differentials and transfer cases that are no longer created when those components are not part of the chosen powertrain type. For example, a rear wheel drive powertrain will no longer show variables from this list pertaining to the front axle or central transfer case.

Table 12. Variables that are not created unless needed for the selected powertrain type.

State Variables	Import Variables	Output Variables
SV_AV_SHAFT_*	IMP_AV_D*	AV_D*
SV_CL_DIFF_LOCKED_*	IMP_CLUTCH_D*	AV_DS_*
SV_CL_DIFF_SLIP_REF_*	IMP_CLT_D*	ClutchD*
SV_CL_DIFF_SPIN_*	IMP_MY_OUT_D*	Clt_D*
SV_CL_2ND_DIFF_LOCKED_*	IMP_M_DIFF_D*	Clutch*
SV_CL_2ND_DIFF_SLIP_REF_*	IMP_M_OUT_D*	M_*
SV_CL_2ND_DIFF_SPIN_*	IMP_ROT_D*	Rot_D*
SV_CL_TWIN_LOCKED_*	IMP_R_EFF_D*	
SV_CL_TWIN_SPIN_*	IMP_R_GEAR_D*	
SV_M_CL2_CAP_*		
SV_M_CL_CAP_*		
SV_M_CL_*_CAP_DYN		
SV_ROT_SHAFT_*		

## Known Issues

1. Deprecated functions for use in VS Commands listed in Tables 10 and 11 in the *Paths and Road Surfaces Help* document generally do not work when VS Terrain is in use. VS Terrain did not exist when they were deprecated, so no changes will be made.
2. Vehicles with trailers require OPT\_INIT\_SPEED=0 when OPT\_INIT\_CONFIG=0. The OPT\_INIT\_CONFIG parameter triggers an initialization of a vehicle's spring deflections when set to 1 and no such initialization when set to 0. The OPT\_INIT\_SPEED parameter determines if the vehicle and wheel speeds are either initialized (= 1) or not initialized (= 0). These parameters may be set with VS Commands or by using the checkboxes on a **Procedure** or **Event** screen. Starting in version 2018.1, for vehicles with trailers, the run will fail with OPT\_INIT\_CONFIG=0 unless OPT\_INIT\_SPEED=0.
3. The built-in electronic stability controller (ESC) handles several cases: oversteer correction (brake outside front wheel), understeer correction (brake inside rear wheel), and rollover protection (brake all wheels). In the case of understeer correction, there is a bug which causes the understeer correction value to be applied to the outside rear wheel instead of the inside rear wheel. This issue is present starting in version 2019.0. VS Commands and import variables may be used to correct this issue.
4. The powertrain parameter OPT\_PT is used to determine the powertrain type. If the value of OPT\_PT is changed after the run starts with an Event, the simulation will fail in 2020.1. The powertrain properties must instead be modified without changing OPT\_PT.
5. Version 2020.1 introduces a new License Feature in support of Multiple Vehicles. Be aware that if you obtain a license for that feature, you must have TruckSim 2020.1 running as the license manager. If you are working with multiple versions with a network license

manager, be sure to launch TruckSim 2020.1 first if you will need the Multiple Vehicles license option.

6. Examples that use external tire models — e.g., MF-Tyre/MF-Swift, COSIN FTire, VS/STI — cannot be run using the command line.

## TruckSim 2020.0 vs TruckSim 2019.1

### Bug Fixes

The following bugs were identified and fixed in this release.

#### *VS Math Model*

1. The tracking of road contact properties applies to three sets of contact points: the tires, the aerodynamic reference points for each sprung mass, and moving objects. An error was found in how the contact is updated for aerodynamic reference points if the vehicle has a trailer. The error was negligible in most cases, but could cause problems in scenarios involving multiple road surfaces and boundaries between them. The parameter `CURRENT_ROAD_ID` is dynamically set to the road ID for reference point of the lead unit, so there were sometimes be an error in this parameter. This issue was introduced in 2018.0
2. The single point driver model now works correctly when reverse gear is specified with an open-loop shift control dataset. Previously, the driver model only looked for `MODE_TRANS_CONSTANT = -1`, which did not account for open loop specification of reverse, which would have `MODE_TRANS_CONSTANT = 1`.
3. Reverse light visibility logic has been updated to better handle minimum powertrains, powertrains without transmissions, and powertrains with manual transmissions. Previously, the reverse light status was established only by the output `ModeTran`. Now, based on keyword `OPT_PT` and the presence of the `INSTALL_TRANSMISSION` command, either `ModeTran` or `GearStat` are used. Minimum powertrains do not support reverse lights, as there is no reverse gear.
4. In the speed controller, the reset of the integral of speed error could occur at the half-step if using a half-step integration method such as AM-2. If this happened, the reset was lost and the integral remained in effect, degrading the performance of the controller.
5. In the speed controller, the acceleration control mode (`OPT_SC = 5`) was disabled if the brakes were applied.
6. In the electric or hybrid-electric powertrain (`OPT_HEV > 0`), spin inertia of engine, electric motor and planetary gear sets were half counted in the previous version. This has been fixed by full counting of those spin inertia.
7. Target objects with shape type segment had an issue involving the visibility, such that the target was not always detected as expected.
8. The Bearing Angle calculation for the ADAS Sensors reported incorrect information in cases where the ADAS Sensor's Yaw Aim Angle was at or near 180 degrees (Revised June 2020 to include this item).

9. Embedded Python did not work for the 64-bit TruckSim solver.
10. The `TLAG_CL_TWIN` parameter (twin clutch time delay) was ineffective and behaved as a zero delay due to a bug in versions 2019.0 through 2020.0. This has been fixed for 2020.1.

### *VS Browser*

1. On the **Events** screen, the checkbox **Reset control clocks** is used to automatically write a number of lines into the Parsfile for the dataset that cause Configurable Function `TSTART` offsets to be set to the current time (e.g., `TSTART_BRAKES`, `TSTART_SPEED_TARGET`, etc.). Those lines of text appeared in the Parsfile before the links to other datasets. For most of the controls, this is OK. However, in recent versions, multiple datasets may be defined for the `SPEED_TARGET` function. The sequence of statements in the Parsfile would reset the `TSTART_SPEED_TARGET` value for the current table, as identified with the system index `ISPEED`. If a blue link is made to a Speed Target dataset, the reset would probably apply to the wrong one.
2. The **Tools Find All References** did not work for **Events** datasets that had the original layout in which a variable of interest is specified in a field, and a threshold value is specified in another. In these cases, the new command `MAKE_EVENT` was written in the Parsfile instead of `DEFINE_EVENT` (which now has a simpler and more powerful syntax).

## **Backward Compatibility**

### *Solver64.exe*

The external solver wrapper `Solver64.exe` has been removed from all products. This tool has been replaced by `VS_SolverWrapper_CLI_64.exe`. There is also a 32-bit version of the Solver Wrapper. More information can be found in the *VS Solver Wrapper* technical memo.

### *Control: Shifting (Closed Loop) screen*

As of version 2019.1, the screen **Control: Shifting (Closed Loop)** was updated such that it was no longer possible to link directly to the **Control: Shifting (Open Loop)** screen using the drop-down menu in the lower left corner. However, if a database or `.cpar` was created in version 2019.0 or earlier that included a link to the **Control: Shifting (Open Loop)** screen from the **Control: Shifting (Closed Loop)** screen and then it is opened with version 2020.0, the **Control: Shifting (Open Loop)** dataset will still be in the screen parsfile for the **Control: Shifting (Closed Loop)** screen. The effect is that the data specified on the **Control: Shifting (Open Loop)** screen will still be used in the simulation.

For version 2020.0, the recommended practice is to break the link to the **Control: Shifting (Open Loop)** dataset by toggling the library link for the Closed-loop shift control settings (i.e., select **[No linked library]**). If the simulation is supposed to use a dataset from the **Control: Shifting (Open Loop)** library, then that link should be made on the **Procedure** or **Event** screen (Revised June 2020 to include this item).

## Known Issue

### *Importing expanded parsfiles from version 8.2.2 will crash the VehicleSim Browser*

When importing an expanded parsfile from version 8.2.2 of the VehicleSim Browser, a check for broken links will be performed prior to the Browser shutting down. We recommend importing parsfiles from version 8.2.2 into an earlier version of the VehicleSim Browser then exporting to a CPAR for import into 2020.0.

## TruckSim 2019.1 vs TruckSim 2019.0

### Bug Fixes

The following bugs were identified and fixed in this release.

#### *VS Terrain*

The VS Math Model was extended to support VS Terrain when using ground information from external sources, and for running in external environments such as EPIC Unreal. The VS Scene Builder tool was extended in 2019.0 to include VS Terrain 3D information for all scenes exported for use in TruckSim, using a new command `VS_TERRAIN_FILE` to load a `.vsterrain` file. However, after shipping 2019.0, we realized the integration was not complete.

The option to use VS Terrain is “all or nothing.” When VS Terrain is in use, VS Roads are disabled. Attempts to add roads after the `VS_TERRAIN_FILE` command has been used should generate an error message. Alternatively, if a VS Road has been defined, an attempt to use the `VS_TERRAIN_FILE` command should generate an error. This was not the case in 2019.0.

When the `VS_TERRAIN_FILE` command was used in a 2019.0 simulation, the information was not written into the Echo file. Also, all information related to VS Roads and associated parameters and Configurable Functions was still shown in the Echo file, even though the VS Roads were not used in the simulation.

All scenes generated with VS Scene Builder were provided with a VS Terrain file, even if the 3D geometry simply specified a flat surface. Some of the examples made in version 2018.1 with pedestrians and running deer used VS Roads to mimic the apparent 3D surface shown in the tiles used in VS Scene Builder. In the 2019.0 version, those VS Roads existed but were not used. In the 2019.1 version, all tiles have true 3D surface properties, so additional roads are not needed.

#### *Scene: External Import screen and VS Terrain*

In TruckSim 2019.0, the screen parsfile for **Scene: External Import** allowed a link to either a `.csv` file (from Atlas, <https://atlas.carsim.com>) or a `.vsscene` file (from Atlas or the VS Scene Builder). When the `.vsscene` file came from the VS Scene Builder, an associated file called `.vsterrain` was included and written into the **Scene: External Import** screen.

A bug was found for the case of a dataset that contained a `.vsterrain` reference. If the dataset was modified by a user to link to a `.vsscene` file downloaded from Atlas, the screen parsfile would retain the link to the `.vsterrain` file as a carryover from the original link to the `.vsscene` file from the VS Scene Builder. As a result, simulations would see the vehicle falling



through the road. In this case, the only solution (in 2019.0) was to open the screen parsfile for the **Scene: External Import** screen, check and see if a link exists to a `.vsterrain` file, and delete that line of text.

If the **Scene: External Import** screen parsfile is not modified to delete the link to the `.vsterrain` file as described above, then following a database conversion from 2019.0 to 2019.1, the database rewrite will not automatically clear the **Scene: External Import** screen parsfile's link to the `.vsterrain` file. In this case, the solution is to browse to the **Scene: External Import** screen and reimport the files downloaded from Atlas (`.csv` or `.vsscene`). This will properly clear the link to the `.vsterrain` file.

### *2019.0 VS Terrain files cannot be read by 64-bit TruckSim VS Solver*

Any `.vsscene` file created by VS Scene Builder in TruckSim 2019.0 created a `.vsterrain` file that was incompatible with the 64-bit VS Solver. This was most apparent when running in a co-simulation with 64-bit MATLAB/Simulink. In these cases, the vehicle would not be located properly on the road. Running with 32-bit MATLAB/Simulink worked correctly. Any `.vsscene` files built using the VS Scene Builder in version 2019.0 will have to be rebuilt for use with 64-bit solvers.

### *Acceleration control of moving objects*

The acceleration control for traffic vehicles that was introduced in 2019.0 includes two parameters — `MAX_AX_BRK_OBJ` and `MAX_AX_THR_OBJ` — that limit acceleration applied during braking and throttle, respectively. Two problems were identified after the release and have since been fixed.

The two parameters are supposed to be indexed, using a hidden parameter `IOBJECT` to define the context when the root keyword is used. For example, the limit in braking acceleration for object #3 can be set using the keyword `MAX_AX_BRK_OBJ(3)`, as it appears in the Echo file, or by first setting `IOBJECT` to 3 and then using the root keyword `MAX_AX_BRK_OBJ`. This did not work in 2019.0; the explicit keyword (e.g., `MAX_AX_BRK_OBJ(3)`) had to be used.

When the acceleration for the object (`A_Obj_o`, where `o` is the object number) is negative, there was no limit; the parameter `MAX_AX_BRK_OBJ` was ignored.

### *Moving objects and ego vehicle elevation*

The elevation of moving objects and the ego vehicle could become erroneous under certain conditions. It was possible to design a road and path such that the initialization of the elevation, based on the calculated road station, would result in an incorrectly calculated station position along the same road path. If the road had any elevation change between the intended station and the incorrectly calculated station, the object or ego vehicle will appear with an incorrect elevation.

This issue was fixed by performing a more thorough sweep of the road path during initialization, to find most appropriate station value given the setup parameters for the simulation. In the case of the ego vehicle, the new calculated value is written to the state variable `SV_STA_ROAD`. For situations where end-users want to disable this behavior, or to manually specify a value for `SV_STA_ROAD`, the parameter `OPT_INIT_STA_ROAD` may be used in any Miscellaneous yellow field. `OPT_INIT_STA_ROAD` can be set to a value of zero (0) to disable the enhanced station search algorithm and allow manual specification of `SV_STA_ROAD`, or to a value of one

(1) to enable the enhanced search algorithm which automatically specifies SV\_STA\_ROAD. The default value for OPT\_INIT\_STA\_ROAD is one.

### *Database conversions did not flag user-defined Path IDs if they were less than 999*

Starting with TruckSim version 2017.0, user-defined IDs for roads, paths, and tables needed to be set to integer values of 999 or larger. Converting a database from version 2016.2 and earlier which used user-defined Path IDs less than 999 were not flagged. The result was Path datasets would be reset to use automatic IDs as determined by their dataset linked order, and VS Commands and Events that referred to these Path IDs in Miscellaneous yellow fields would no longer be valid. Starting in TruckSim 2019.1, all screens with user-defined IDs are subject to the same rules: symbols entered into the fields (i.e., a user-defined parameter) are ok, while user-specified integers must be 999 or larger. Integer values specified to be 1 – 998 will generate an error message.

### *GPS coordinates*

TruckSim supports the import of GPS coordinates for roads and the output of GPS coordinates for vehicles. The calculation method has been unified for both cases and is described in the technical memo “GPS Coordinates,” which replaces the memo “GPS Coordinate Outputs”. Previously, GPS import used the haversine formula to convert latitude and longitude to X and Y. As of 2019.1, GPS import uses the same method as the output variable calculation, which is based on the WGS 84 reference ellipsoid. Additionally, when a GPS path is imported, the reference point is now reset according to GPS\_RANGE\_Y, as done for the GPS output variables GPS\_LatA and GPSLongA.

### *Hybrid powertrain*

The hybrid powertrain received several major improvements since being introduced as a built-in option in version 2019.0.

1. The option to select “Reverse” using the Closed Loop Shift Controller was added.
2. The output variable Thr\_HEV was hard-coded to report 0.9. This has been changed such that it is modulated based on the ratio of the required engine torque and the full-throttle engine torque.
3. In some cases, the vehicle could drive in reverse such as when the initial battery charge was set to zero or if the data used for the hybrid system resulted in a reversal of the spin direction of the planetary gear set due to torque imbalance. The model has been updated to more aggressively check for negative motor torque and limit it regardless of the engine speed.

### *Miscellaneous bug fixes*

1. The screen **ADAS Sensors for Range and Tracking** has a yellow field to set the parameter SENSOR\_N\_DETECT, which is only shown when the box “Show all detection lines” is unchecked. In some cases, setting this parameter to a value that corresponded to a sensor / moving object combination that did not exist (e.g., 15 Moving Objects, but SENSOR\_N\_DETECT was set to 16) would write invalid animator reference frame data.
2. Some settings for ADAS Sensors made from the Sensors screen did not work for sensors numbered 27 and higher.

3. Loading animator resources from complicated ADAS examples with tens of thousands of detection variables could take long enough that the software would appear to be unresponsive and/or crashed. The loading process was improved substantially, with progress being shown to avoid the appearance of being unresponsive or otherwise crashed.
4. The screen **Multiple Moving Objects** was set up to read the Miscellaneous yellow field, located at the top of the screen, after the yellow data field for the moving object's wheelbase, which is part of the off-tracking feature. This could cause problems with advanced setups where the sequence of reading the contents was critical (e.g., using formulas involving other parameters).
5. The Moving Object output variable `V_Obj` was not calculated when the object is located using the option "Formula for Station." The error was in the Parsfile written by the Browser. This affects version 2019.0 only.
6. The screen **Control: Shifting (Open Loop)** has a link that is intended to include a Clutch Control dataset. There was a bug involving the screen refresh that prevented the link from being made.
7. The Transmission option to lockup the torque converter could not be used when the transmission shifting control was set to Open-Loop.
8. Running a simulation with a rear-wheel drive (RWD) powertrain wrote Twin Clutch data into the Echo file even though a Twin Clutch differential was not specified. This did not occur for front-wheel drive (FWD) and four-wheel drive (4WD) powertrain configurations.
9. The screen **Suspension: Spring** could not start with a 1D table and be changed to a 2D table. This affected version 2019.0 only.
10. The screen **Suspension: Compliance (Nonlinear)** incorrectly wrote a math model keyword for the compliance Dive/MyBk, affecting Independent suspension settings. The effect was the value for the linear coefficient was ignored when set directly on the **Suspension: Compliance (Nonlinear)** screen. The parameter was written as `CD_MZ_COEFFICIENT`; it should be `CD_MY_COEFFICIENT`.
11. The screen **Animator: HUD** had an error involving the option to select an arbitrary output variable to control the speedometer needle: the screen parsfile was supposed to include `WRT_` and `ANI_` commands to make the selected variable available to VS Visualizer. Without those commands, the variable would not be written to the output file unless it was also specified for plotting, or if all variables were written to file (i.e., Write all outputs). If the variable was not written to the output file, the speedometer needle would not move.
12. Cloning a dataset did not clean up all broken links. The option to rename a category or dataset (**Edit > Change Title or Category of This Dataset**) presents three small checkboxes allowing an end-user to clone the dataset, redirect links from the original dataset the clone, and delete the original dataset. In some cases, not all links were cleaned up, resulting in broken links.
13. A run would stop immediately if the parameter `ROLL_MAX_STOP` was set to a negative value. The solver now compares the absolute value of the roll angle with the absolute value of `ROLL_MAX_STOP`.

14. TruckSim could crash if a Batch Run was performed with an empty Run Control dataset. This was an uncommon use-case and occurred when a new, empty database was created, after which a Batch Run was initiated.
15. Sorting columns of data on the Import Preview window resulted in the import process failing. When importing a .cpar file, an option exists to preview the importing data allowing end-users to see not only which individual parsfiles are being imported, but also whether or not those importing parsfiles will conflict with ones that already exist in the database. Clicking on the column headers sorts the columns and was intended to accommodate situations in which end-users wanted to quickly prioritize what was being shown rather than scrolling down a potentially lengthy list. Unfortunately, this sorting had the unintended side-effect of the parsfiles failing to import.
16. The indexed parameter `IT_XXZZ` was added for the XX / ZZ moments of inertia of the tire assembly in version 2019.0. However, its value was incorrectly overwritten by the wheel mass of an external tire model when a 3<sup>rd</sup>-party tire model (e.g. MF-Tyre/MF-Swift, COSIN FTire, etc.) was used.
17. Wheel Center Force Arrows did not display in the correct location, starting in version 2017.0, in which the Wheel Center Force Arrows were being shown at the Center of Tire Contact (CTC) instead of the wheel center.
18. The VS Command function `ROAD_DZDS_ID` and corresponding API function `vs_road_dzds_id` that both calculate road slope  $dZ/dS$  included only the 2D `ROAD_DZ` tables; they omitted the effect of the 1D `ROAD_ZS` table.
19. VS Command equations with more than 300 characters would not run in real-time on dSPACE. The internal limit has been increased to 512, and error handling has been improved.
20. When using the VS Connect API function `UpdateData_CopyData()` for TruckSim / Unreal Engine simulations, it did not copy data and failed without any error messages. This problem only appeared when trying to use the API function; it was not used in any of our shipping TruckSim Unreal Engine examples.
21. The TruckSim Lead Unit screens (2 and 3 axles) and the TruckSim Vehicle Assembly screens have a miscellaneous link that was written in the dataset Parsfile before the Powertrain link. In 2019.0 the powertrain was not installed until the dataset was loaded, which meant a linked miscellaneous dataset could not reference parameters or output variables for the powertrain.
22. The TruckSim screen **Vehicle: Trailer with 4 Axles** had a checkbox to install the speed controller. This had no effect but was misleading.

## Backward Compatibility

### *Change of location for external examples*

A number of examples that were previously bundled with the VS Browser up until 2019.1 have been moved to the new VS SDK. These examples include the ones related to the VS API and other

libraries relevant to developing custom applications built around VehicleSim products. In order to obtain the SDK and any of these examples please visit the Mechanical Simulation website.

### *Examples made with roads from VS Scene Builder*

VS Scene Builder was introduced in 2018.0 and subsequently upgraded in 2018.1, 2019.0 and 2019.1. Scenes imported into TruckSim from 2018.0 and 2018.1 had paths and animator assets, but no 3D terrain information. VS Scenes imported in 2019.0 and 2019.1 also include 3D terrain information using a VS Terrain file (extension `.vsterrain`). If examples prior to 2019.0 are loaded and exported using 2019.0 or newer, they will have VS Terrain data. As noted above, errors will be generated if an attempt is made to also define VS Roads, as was often done in examples prior to 2019.0.

### *Driver model for closed-loop steering*

The steering controller was modified for version 2019.1 to include the capability of driving backwards. In reviewing the software, several changes were made in support of the backward travel mode and for situations in which the angle between the vehicle yaw and the target path is significant. In many cases, there is little difference in performance; in some cases, the older version might be preferable. For version 2019.1 a new parameter has been added: `OPT_DM_2019`. When set to zero (the default), the new version is used. When there is a need to reproduce results from older versions of TruckSim, set this parameter to one.

### *Legacy Simple Suspension screens*

TruckSim originally had suspension screens for both kinematics and compliance data, when the suspension models were symmetric and did not have bump stops. The libraries were named **Suspensions: Independent System (Legacy Simple)** and **Suspensions: Solid Axle System (Legacy Simple)**. TruckSim has not included any example datasets in libraries library for years. Both screens had red text advising that they should not be used. Both screens are now gone.

### *Powertrain Import variables for differentials*

The efforts aimed at modularizing the powertrain mean that powertrain import variables cannot be used unless their associated modules are installed. For example, if the vehicle has a 2WD powertrain (a single differential) but the import variables for the clutch control of the transfer case are in the I/O Channels: Import array (referring to the third differential), a warning is issued when the simulation is run indicating that there are invalid variable names. Therefore, only import variables corresponding to the current powertrain configuration should be in the I/O Channels: Import array, and all others will need to be removed.

The imports for differential output shaft torque have been removed for the “minimal” powertrain option (`OPT_PT = 0`). In previous versions, it was possible to import the output shaft torque for differentials that were not in use due to the minimal powertrain. For example, “Differential 2” output shaft torque would be applied to the rear wheels, in the same manner as for the detailed RWD powertrain configuration. Moving forward, wheel torque imports such as `IMP_MYSM_L2` or `IMP_MYUSM_L2` should be used instead. (`IMP_MYSM_L2` is reacted at the sprung mass, `IMP_MYUSM_L2` is reacted at the unsprung mass.)

### *Running simulations from the command line on Linux*

The symbolic link used to launch TruckSim runs from the command line on Linux has been renamed from “trucksim” to “trucksim-cli”, to remove any confusion with the VehicleSim Browser on windows, which is named “trucksim.exe“. Any scripts or workflows created by the end-user which rely on the “trucksim” command will need to be updated.

### *Using an Existing Simulink model on ETAS LabCar*

For running TruckSim on ETAS LabCar, if the user wants to use an existing Simulink model and upgrade to TruckSim 2019.1, it is necessary to replace the existing TruckSim S-Function Block with the one in the Library Browser and disable the link.

## **Known Issues**

### *Steering system compliance remains in effect when OPT\_STEER\_EXT = 4*

When OPT\_STEER\_EXT is set to 4, the intention is that the TruckSim Steering System is fully replaced. The Steering System Compliance (as linked on the Steering System screen), however, is still included in the model. End-users can disable the Steering System Compliance by setting it to zero using the compliance link on the Steering System screen or in a Miscellaneous yellow field. Long-term, the intention is to disable the effects of the Steering System Compliance as set on the Steering System screen when OPT\_STEER\_EXT = 4.

### *VS API access for optional components*

In some VS API code examples, the `setdef` callback function is used to access internal variables. For optional vehicle components (e.g. powertrain), associated internal variables cannot be accessed from the `setdef` callback because these variables are defined later in the initialization process. The recommended course of action is to access the affected variables only after explicitly calling `vs_setdef_and_read` (Figure 2).

```
vsDLL = vs_load_library(pathDLL);
vs_get_api(vsDLL, pathDLL);
vs_setdef_and_read(simfile, NULL, NULL);

/*-- Variables for powertrain can be accessed here --*/
vs_statement("IMPORT", "IMP_THROTTLE_ENGINE vs_replace", 1);
sImpStr = vs_get_var_ptr("IMP_THROTTLE_ENGINE");
/*-----*/

vs_initialize(t, NULL, NULL);
while (!vs_stop_run()) {
    vs_integrate(&t, NULL);
}
```

Figure 2. C code showing `vs_statement` used for VS Command.

# TruckSim 2019.0 vs TruckSim 2018.1

## Bug Fixes

The following bugs were identified and fixed in this release.

### *VS Math Model*

1. Tandem and Tridem Load Sharing suspension groups include a coefficient to define the load transfer in proportion to the total brake torque applied to all wheels of that load sharing group (math model keyword `CFZ_MY_TAND`).

In TruckSim 2018.0 and 2018.1, these coefficients were used on the lead vehicle unit (i.e., vehicle unit 1) but were inadvertently disabled for trailers. This has been fixed for TruckSim 2019.0.

2. In 2018.1 and earlier, the derivatives of two state variables for the built-in tire model were not being correctly handled when the tire left the ground (`SV_SLIP_LONG_tire` and `SV_SLIP_LAT_tire`).
3. The ADAS Sensor model was extended in 2018.0 to use a new algorithm for determining occlusion among target objects. The option was installed to use the older algorithm by using a parameter `OPT_LEGACY_SENSOR` that could be set for each sensor. As with other indexed parameters, it should be possible to identify this parameter with two methods:
  - a. use custom keywords for each sensor (e.g., `OPT_LEGACY_SENSOR(1)`, `OPT_LEGACY_SENSOR(2)`, etc.), or
  - b. use the root keyword `OPT_LEGACY_SENSOR`, with the context defined with a system index parameter `ISENSOR`.

The second option did not work in versions 2018.0 and 2018.1.

4. In versions 2018.0 and 2018.1, the pitch and roll angles of moving objects located on a road surface were wrong if the path used to locate the object was the same path used as reference for the road. The error did not occur if the road was flat, or if the path used for the object was different than road reference path, or if the road was straight and heading east (such that  $S = X$ ,  $L = Y$ ).
5. The orientation of the axes for a motion sensor are adjusted by three rotation parameters `A_ROLL_S`, `A_PITCH_S`, and `A_YAW_S`. If two or more of the angular adjustments are nonzero, then the orientation of the axes depends on the sequence of rotations. Starting with the sensor axes parallel with those of the vehicle part, the adjustment is first made by the specified roll angle about the vehicle part X axis, then the pitch angle about the new Y axis of the sensor after the roll adjustment, then the yaw angle about the new sensor Z axis.

The roll sequence has been consistent ever since motion sensors were introduced many years ago. However, the documentation for the motion sensor screen incorrectly stated the wrong rotation sequence in versions 9.0.3 through 2018.1.

6. TruckSim includes 24 built-in forces and moments acting on each unsprung mass either from the sprung mass or from the inertial reference (ground). These all have default magnitudes of zero, but can be assigned values by some advanced users working on custom suspension configurations. These included  $IMP\_F(x,y,z)EX\_ (whl)$ ,  $IMP\_F(x,y,z)GEX\_ (whl)$ ,  $IMP\_M(x,y,z)EX\_ (whl)$ , and  $IMP\_M(x,y,z)GEX\_ (whl)$ .

The actions of these user-defined forces and moments were accidentally removed in the conversion to a modular design in version 2018.0 and were still missing in 2018.1.

7. A small error was made in the 3D kingpin steering geometry for solid axles in TruckSim 2018.0 when the older approximate model was replaced with the more detailed model from TruckSim. An offset exists for kingpin axes with inclination angles. The offset is very small but can cause a gradual drift in direction with zero steering angle, even if all compliances are zero. The geometry was handled correctly with independent suspensions, so the effect could only be seen for vehicles with solid axles in front.
8. Moving objects can be set to follow a path using a parameter  $PATH\_ID\_OBJ$  that identifies the ID number of a path, with an offset calculated by the  $LTARG$  function using a dataset identified with the parameter  $LTARG\_ID\_OBJ$ . If these parameters were changed during an Event, the links to new datasets for the path and/or  $LTARG$  function were not applied unless the Event triggered a new initialization.

The parameters have been redefined such that they trigger an initialization if changed during an Event. (This is indicated by the code [ I ] shows at the end of the parameter descriptions in the Echo file.) A workaround is available for 2018.1 (see Known Issues for 2018.1, page 49).

9. Using VS Commands to call any table function with interpolation type  $STEP$  would return zero at all times in version 2018.1. A workaround is available for 2018.1 (see Known Issues for 2018.1, page 49).
10. TruckSim would crash when an undefined variable was used in a formula specified in an Event when reading data for a Configurable Function. Error handling has been improved so run stops cleanly after the error message appears.
11. TruckSim would crash when using a table index of 2 or greater while also specifying the table index of 2 in an equation.
12. When resuming a simulation using the End Parsfile from an existing run, there would sometimes be an error locating the aerodynamic reference points on a road surface.
13. Road friction is set to 0 for some special cases, such as emulating a K&C test setup. In some cases, the solver crashed. A workaround is available for 2018.1 (see Known Issues for 2018.1, page 49).
14. In all older versions of TruckSim, changing a differential gear ratio for 2WD and 4WD powertrains during a run could produce a large torque spike.
15. In all older versions of TruckSim, adding an output variable during an Event could cause a crash if the option to write all outputs was active. Trying to add an output variable during an Event will now generate an error.



16. In version 2018.1, the VS Browser could crash during the import of an encrypted expanded parsfile. This has been fixed for version 2019.0.

### *Model screens*

1. Linking to new datasets failed sometimes for some blue links on the **Procedures** library screen. Links with ID numbers #27 through #33 would fail to write to the Parsfile properly in rare scenarios when certain conditions were met.
2. The GUI screen **ADAS Sensors and Moving Objects > Single Moving Object (Custom)** has options for starting and stopping the motion of a target object based on conditions that are specified with formulas. Those formulas were written to the Parsfile for the dataset using the EQ\_OUT command. If the integration method (specified with the parameter OPT\_INT\_METHOD) used a half-step (RK-2 or any of the AM methods), inconsistent outputs could be generated if the start or stop conditions changed values at a half-step. The Parsfiles now use the command EQ\_FULL\_STEP such that any ambiguities are eliminated.
3. Some of the suspension screens have an option for setting the Wheel Center Height for both wheels at that axle position. If the box is checked to show the data field, data is entered and will be used. A bug in most past versions going back to 8.2 is that if the box is unchecked, hiding the data field, they would still be written to the Parsfile dataset.

This behavior was incorrect; options that are unchecked are not supposed to be used by the math model.

## **Backward Compatibility**

### *Optional Closed-loop speed controller*

Two improvements have been made to the built-in closed-loop speed controller.

The controller has been made optional; it does not exist unless installed by the command `INSTALL_SPEED_CONTROLLER`. The installation command is applied automatically by the GUI screens if it is used at the start of a simulation. However, if a simulation begins using open-loop throttle and no speed control, and later switches to speed control using an Event, an error will be reported because the `INSTALL_SPEED_CONTROLLER` cannot be used after the run starts (this is because it adds state variables to the math model). In this case, you must insert the command in a miscellaneous field, usually in the **Procedure** dataset used for the run.

The controller uses integral feedback to match target speeds closely. A new option was added to reset the integral of the speed error when the speed error changes sign (crosses zero) and the accumulated error is greater than a specified “dead zone” threshold. The new options work well in most situations, and are now enabled by default. Differences in simulation results can be expected when the target speed changes significantly, such as speeding up to a new speed limit, are dropping to make a tight turn. In these situations, the new controller may behave differently. On the other hand, when maintaining a constant speed on a flat surface, the behavior should match earlier versions.

### *Powertrain Variables*

Several powertrain state variables and output variables have been reorganized such that their naming matches that of other per-wheel variables, starting at “1” and counting backward along the vehicle, per axle.

L1, R1, L2, R2, L3, R3...

In previous versions, the naming arrangement was based on an internal structure specified per powertrain type, such as RWD, AWD, 6WD, etc. For example, for a 5-axle vehicle with a FWD powertrain, the solver would write outputs to the “L1” and “R1” variables, despite axle 4 being powered. The new naming would reflect the powered axle wheels, “L4” and “R4”.

Variables for unpowered wheels will have zero values.

List of affected state variables:

SV\_ROT\_SHAFT\_L1, et al  
SV\_AV\_SHAFT\_L1, et al

List of affected output variables:

MY\_DR\_L1, et al  
MY\_DR\_L1, et al

### *Optional Powertrain Components*

Several individual powertrain components have been made optional. The following new commands have been added to install specific components:

INSTALL\_ENGINE  
INSTALL\_TRANSMISSION  
INSTALL\_TORQUE\_TRANSFER\_DEVICE  
OPT\_PT (Required for all powertrains)

These commands will install the necessary state variables, parameters, and outputs. The powertrain GUI screens add these commands automatically, but in some cases, you will need to add an installation command to a miscellaneous field.

OPT\_PT is normally automatically written by the powertrain GUI, but if your particular run data omits OPT\_PT, then associated powertrain state variables, parameters, and input/output variables will not be installed.

For example, the following example will result in an import variable error if there is no actual engine installed, or if the engine is installed later in the parse file sequence.

DEFINE\_OUTPUT TorqNorm = M\_Eng\_In / 4.5

Precede the command with the INSTALL\_ENGINE command to access the M\_ENG\_IN variable.

INSTALL\_ENGINE  
DEFINE\_OUTPUT TorqNorm = M\_Eng\_In / 4.5

Note that for external powertrain components (e.g. external engine modeled in Simulink), it is still necessary to install the component in question, or else necessary import variables will not exist.

### Simulink S-Function Initialization

Along with changes for earlier initialization, S-Functions now run a full time step at  $T = 0$ . In earlier versions, this initial timestep was skipped and not calculated. Extra care must be taken to ensure that valid data is supplied to the S-Function at  $T = 0$ , or else the model may experience instabilities in 2019.0.

As specific example, your Simulink model might use a memory block feeding into the TruckSim S-Function import variables. The memory block will supply zero values by default for that first timestep. Zero values for some variables, such as gear ratios, will cause the model to fail. The appropriate fix in this case is to remove the memory block or configure it to supply valid initial values.

### State variables

Testing of the internal calculations made during the `LINEARIZE` command identified several variables that were not restored properly during perturbations. These also caused some minor discontinuities when resuming a simulation using an End Parsfiles from an earlier run. New state variables were added as needed, and a few existing state variables were renamed for consistency (Table 13).

Table 13. Changes in names of state variables.

Old Keyword	New Keyword	Index
<i>new</i>	SV_CMP_T_t	$t=LF, LR, RF, RR$
<i>new</i>	SV_DRGEAR_CVT	
<i>new</i>	SV_D_ALPHA_t	$t=LF, LR, RF, RR$
<i>new</i>	SV_D_ALPHA_MX_t	$t=LF, LR, RF, RR$
<i>new</i>	SV_D_CONI_t	$t=LF, LR, RF, RR$
<i>new</i>	SV_D_KAPPA_t	$t=LF, LR, RF, RR$
<i>new</i>	SV_GEAR	
SV_GEAR	SV_GEAR_STAT	
<i>new</i>	SV_LRELAX_X_PAC52_t	$t=LF, LR, RF, RR$
<i>new</i>	SV_LRELAX_Y_PAC52_t	$t=LF, LR, RF, RR$
<i>new</i>	SV_M_TC	
<i>new</i>	SV_SPEED_DVDS	
<i>new</i>	SV_STR_DM	
SV_STR_DM	SV_STR_DM_OLD	
<i>new</i>	SV_STR_KP_w	$w=LF, LR, RF, RR$
SV_STR_KP_w	SV_STR_KP_w_OLD	$w=LF, LR, RF, RR$
<i>new</i>	SV_V_O_i	$i=1,2,3...$
<i>new</i>	SV_VERR_OLD	
<i>new</i>	SV_Z_CTC_t	$t=LF, LR, RF, RR$

### *Parameters and the Echo file*

Some parameters were renamed and/or moved to new sections of the Echo file (Table 14).

*Table 14. Keywords changed for 2019.0.*

<b>Old Keyword</b>	<b>New Keyword</b>	<b>Old Echo File Section</b>	<b>New Echo File Section</b>
PMAX_SC	PMAX_SC	Speed Controller	Powertrain
R_REAR_DRIVE_SC	R_REAR_DRIVE_SC		
OPT_RECT_OBJ	OPT_SHAPE_OBJ	Moving Objects	Moving Objects

### *Links to Audio Datasets*

Years ago, TruckSim databases included animator vehicle datasets with links to sound sets, which supported audio in VS Visualizer for wind, engine, and tire audio. When moving objects were added to the models that could be used to mimic traffic vehicles, there could be problems with audio data from a traffic vehicle overriding the data for the ego vehicle. To avoid this problem, the recommended practice for adding audio to VS Visualizers videos has been to link tire sound sets to the Tire dataset, wind sound sets to the Aerodynamics dataset, and engine noise sound sets to the Engine dataset. However, the links to sound sets were left on the animator vehicle screens in support of legacy datasets that were imported.

Those links have been removed from 2019.0. The reason is that the powertrain model is no longer built in, and a result is that sound sets with VS Command equations that refer to engine variables will generate errors if the engine has not been installed.

Because the sound set links were removed from the animator screens, old datasets that added sound sets linked to animation screens will not have sound when imported into 2019.0. The sounds sets are imported and exist in the updated database, but will not be used because the old links are gone. To restore the sound, link to the sound set datasets from the appropriate vehicle component screen: Engine, Tire, and Aerodynamics.

### *Discontinuing Support for dSPACE DS1005 and DS1103*

With the VehicleSim 2019.0 product release, dSPACE DS1005 and DS1103 boards will no longer be supported. The solvers for DS1005 and DS1103 will continue to be built and delivered. However, technical support will not be provided for their installation and use. dSPACE discontinued shipping of the DS1005 and DS1103 in December 2016. dSPACE will cease support for the DS1005 and DS1103 in December 2020 and December 2019, respectively.

### *Shift 2WD to 4WD Example Upgrade Issue*

For older databases that have been upgraded to 2019.0, the triggering condition for the shift to 4WD in the “Shift 2WD to 4WD” example might not be reached, and the vehicle could fail to traverse the course. The example has been corrected in the supplied example data for the 2019.0 product.

## Known Issues

### *AVL Cruise, Locked Brakes issue*

There exists a known issue involving braking. When using the AVL Cruise external powertrain, and when the wheel speeds approach zero while braking (locked brakes), the vehicle will be subject to undue pitching due to incorrect braking moment being applied to the suspension. (The problem is due to an incompatible handling of wheel lockup in braking by TruckSim and AVL Cruise.)

## TruckSim 2018.1 vs TruckSim 2018.0

### Bug Fixes

1. The speed controller parameter BK\_PERF\_SC has units of g/MPa and the output variable Ax\_SCrq has units of g. The automatic conversion of units was inconsistent with internal code, resulting in an error by a factor of g (9.80665 m/s<sup>2</sup>) when converting to user units. The error involving Ax\_SCrq affected plots made with this variable, but did not affect the speed controller. However, the error involving BK\_PERF\_SC did affect behavior during braking. The value used internally was too high by a factor of 9.80665. When working with old datasets, values specified for BK\_PERF\_SC can be multiplied by 9.80665 in order to obtain the same controller behavior.
2. The 2018.0 Beta math model had dynamic spring/damper connections between suspensions and sprung masses. A better connection was used in the release version; however, output variables for the Beta X and Y deflections and forces were inadvertently left in the solver (the values were always 0.0). These outputs were removed for 2018.1.
3. One of the output variables installed by the VS Command INSTALL\_DRIVER\_PREVIEW\_SENSORS was renamed for consistency with other outputs. The old name was X\_RdS\_i; the new name is X\_Targi. The Y variable was also supposed to be renamed, but there was a mistake. The Y variable is now defined properly with the name Y\_Targi.
4. When running with external software such as Simulink, the Model library screen has an option for setting the simulation time step. If that option is not used, the settings from the most recently viewed **Preferences** dataset are supposed to be sent to the external software. Instead, a default of 0.001s was sent (the value from the **Preferences** dataset was ignored).
5. The timing for applying powertrain torques to the wheels was delayed by one time step, capable of causing an instability.
6. The **Animator: Camera Setup** screen has a blue link to an **Animator: Reference Frame** dataset that defines the possible motions of an external camera location and look point. If the name or category of the **Animator: Reference Frame** dataset is changed, the new name appears in the blue link for the **Animator: Camera Setup** dataset. However, the dataset Parsfile still contained the old name in some commands sent to VS Visualizer when viewing a video using the **Animator: Camera Setup** dataset. A workaround with older

versions is to use the menu command **Tools>Re-Write this Library** for the **Animator: Camera Setup** library. This is no longer needed in 2018.1.

7. The `LINEARIZE` command did not work with state variables associated with movements of a solid axle in a solid-axle suspension.
8. The output variable `BetaRd` is supposed to be vehicle sideslip normal to the road surface; instead it was the same as `Beta`, the sideslip normal to gravity. The same was true for the rate, `BetaRdR`.
9. The road ID for a moving object did not always change when the object crossed a boundary.
10. The parameter `L` was ignored in VS Command functions `PATH_CURV_ID` and `ROAD_CURV_ID`.
11. The lateral compliance for a solid-axle suspension was not used. It has been restored. As noted earlier, the X and Y compliance deflections are now available as output variables.
12. VS Math Models support up to 99 ADAS range and tracking sensors that detect moving objects as targets. However, if more than 20 sensors are defined using the GUI, the simulation fails with the following message: “ERROR: Too many sensors. (the limit is 20).” The error does not occur if the sensors are defined using VS commands. Up to 99 sensors can be defined using VS commands. The simulation will also run with a combination of up to 19 sensors defined using the GUI and up to 80 sensors defined using VS commands. This bug was fixed in version 2018.1.
13. A bug introduced in TruckSim 2018.0 involving import variables for user-defined forces and moments applied to the suspensions from the sprung mass was corrected. The bug caused these to be properly defined and imported, and their values could be plotted but they were not applied. These included `IMP_F(x,y,z)EX_(whl)`, `IMP_F(x,y,z)GEX_(whl)`, `IMP_M(x,y,z)EX_(whl)`, and `IMP_M(x,y,z)GEX_(whl)`.

## Backward Compatibility

### *Closed-loop speed control*

A bug was found and corrected involving the coefficient `BK_PERF_SC`, used to characterize system-level braking in the closed-loop speed controller. This internal calculation involving this parameter used the wrong units, with the effect that the actual coefficient was larger than the specified value by  $g$  ( $9.80665 \text{ m/s}^2$ ). All values in the 2018.1 database were multiplied by 10 to provide similar behavior.

### *Path: Segment Builder*

Prior to 2018.1, if the radius or curvature segment is negative to specify a right turn, the arc length must also be specified as negative. For 2018.1, users no longer need to enter a negative value.

When importing 2018.0 database to 2018.1, this negative value is not automatically converted, and an error will be displayed on the Path: Segment Builder screen. Users will need to manually edit negative arc length values to a positive value.

## *Model screens*

All the Model GUI screens (**Models: Simulink**, **Models: Self-Contained Solvers**, etc.) have improved logic involving what parameters are written. They also have a new checkbox, and new logic for the box used to show time steps and integration methods. Therefore, the 2018.1 version has a database version 2018.1. When opening a 2018.0 database, it is automatically converted.

In past versions, a checkbox was used to control visibility only for the time step information; the drop-down control for the integration method was always visible. In the new version, the visibility of the integration method and the time step information are both controlled by the checkbox. It is possible that some existing datasets specified the integration method but not the time step. In these cases, the dataset will not show either, and the default settings will be used.

## *Initialization*

Because of the improved initialization process, some transients due to compliance effects jumping from zero to an equilibrium have been eliminated.

When working with external software that connects with Import and Export arrays, the coordination has been improved. This also eliminates some possible transient behavior at the first time step in past versions that was due to import variables not being available.

## *Error checking*

In general, each new release has more rigorous error checking. Inputs and commands that had invalid settings might have been ignored in past versions. Now, they are more likely to be identified with error message in pop-up windows (if enabled) and in the log file.

In this release, attempts to set units of Configurable Functions with the VS Commands `SET_UNITS_TABLE`, `SET_UNITS_TABLE_COL`, and `SET_UNITS_TABLE_ROW` will now generate error messages if the keywords for the table function or units were not recognized.

## *State variables*

As documented in the *VS Solver Reference Manual*, variables in the model that are needed to define the state of a VS Math Model and which normally change during the simulation run are called *state variables*. These variables all have keywords that begin with the prefix “SV\_” and are listed in an Echo file made at the end of each run (the file has the suffix `_End.par`). The list can also be generated and viewed using the **View** button in the lower-right corner of the **Run Control** screen. Some of the state variables are calculated by numerically integrating a derivative defined with an ordinary differential equations (ODE). Those state variables are identified with an “ODE:” at the start of the description. If there is no ODE prefix, then the state variable is an internal variable that must be saved for use in the next time step, or to support advanced options such as linearization.

Some of the keywords used to identify state variables were changed (Table 15). Most of the changes involved non-ODE state variables, which are rarely (if ever) used in VS Commands or set by users; they mainly exist to support restarting a simulation using the `End.par` file from a previous simulation.

On the other hand, a few ODE state variables were renamed. In these cases, both the old and new names are recognized by the solver and can therefore be used in VS Commands or to set initial conditions. However, only the new names are listed in documentation and the End Parsfile.

Some variables identified as non-ODE state variables are no longer needed to define the model state because of recent changes in architecture. These were removed, as indicated in the table.

Five variables in the steering system that were identified as ODE state variables are not actually state variables in 2018.0 and newer (they are just internal intermediate variables). The keywords SV\_ROT\_END, SV\_POS\_GR\_Aa are SV\_DPOS\_GR\_Aa are no longer recognized. These names were used with the LINEAR\_SV command in some linearization examples for axle 1. If old datasets with these commands are imported into v2018.1, those examples will not work until the two commands are removed.

### *Parameters and the Echo file*

Some parameters were renamed for compatibility with other names (Table 16). Both the new names and old names are recognized by the solver; however, only the new names are listed in documentation and the Echo and End files. Some were also moved to other sections of the Echo file.

*Table 15. Changes in names of state variables.*

Old Keyword	New Keyword	Index	ODE?
SV_2ND_CL_DIFF_LOCKED_ <i>i</i>	SV_CL_2ND_DIFF_LOCKED_ <i>i</i>	<i>i</i> = 1,2,3	
SV_2ND_CL_DIFF_SLIP_REF_ <i>i</i>	SV_CL_2ND_DIFF_LOCKED_ <i>i</i>		
SV_2ND_CL_DIFF_SPIN_OLD_ <i>i</i>	SV_CL_2ND_DIFF_SPIN_ <i>i</i>		
SV_AT_LOCKED_STATE	SV_AT_LOCKED		
SV_A_COL	SV_SWA		Yes
SV_A_COL_OLD	SV_SWA_OLD		
SV_A_IN_GR <i>i</i>	SV_STR_IN_GR_A <i>i</i>	<i>i</i> = 1,2	Yes
SV_BETA_OLD	SV_BETA		
SV_BETA_RD_OLD	SV_BETA_RD		
SV_BRAKE_STATE_OLD	SV_BRAKE_STATE		
SV_CL_DIFF_SPIN_OLD_ <i>i</i>	SV_CL_DIFF_SPIN_ <i>i</i>		
SV_CL_TWIN_ <i>w</i> _LOCKED_STATE	SV_CL_TWIN_LOCKED_ <i>w</i>	<i>w</i> =LF, LR, RF, RR	
SV_CL_TWIN_ <i>w</i> _SLIP_REF	SV_CL_SLIP_REF_ <i>w</i>		
SV_CL_TWIN_ <i>w</i> _SPIN_OLD	SV_CL_TWIN_SPIN_ <i>w</i>		
SV_D2A_IN_GR <i>i</i>	<i>removed</i>	<i>i</i> = 1,2	
SV_DAV_CL_OLD	SV_AV_CL		
SV_DA_COL	SV_AV_SW		Yes
SV_DA_IN_GR <i>i</i>	SV_STR_IN_DGR_A <i>i</i>		Yes
SV_DPOS_GR_A <i>i</i>	<i>removed</i>	<i>i</i> = 1,2	Yes
SV_DROT_CL_LK	<i>removed</i>		
SV_DZDX_GND_OLD	SV_DZDX_GND		
SV_DZDY_GND_OLD	SV_DZDY_GND		
SV_ENGINE_STATUS	<i>removed</i>		



Table 15. Changes in names of state variables.

Old Keyword	New Keyword	Index	ODE?
SV_ESC_STATE_OLD	SV_ESC_STATE		
SV_ESC_STATE_OLD	SV_ESC_STATE		
SV_LAT_AXLE_CMP_i	removed	i = 1,2	
SV_LOCKED_CLUTCH_NEUTRAL_OLD	SV_LOCKED_CLUTCH_NEUTRAL		
SV_LONG_AXLE_CMP_i	removed	i = 1,2	
SV_M_HYS_GEAR_i	SV_M_HYS_GEAR_Ai	i = 1,2	
SV_POS_GEAR_Ai	removed	i = 1,2	
SV_POS_GR_Ai	removed	i = 1,2	Yes
SV_R_GEAR	removed		
SV_ROT_ENG	removed		Yes
SV_STR_SUS_F	removed		
SV_STR_SUS_R	removed		
SV_STR_SW_DM	SV_SWA_DM		
SV_STR_SW_OL	SV_SWA_OL		
SV_THROTTLE	SV_THR_DEL		
SV_X_GND_OLD	SV_X_GND		
SV_Y_GND_OLD	SV_Y_GND		
SV_Z_GND_OLD	SV_Z_GND		

Table 16. Keywords changed for 2018.1.

Old Keyword	New Keyword	Old Echo File Section	New Echo File Section
LTARG_DM	LTARG_ID_DM	Reference Paths	Driver Model: Steering
N_LTARG	N_LTARG		
PATH_ID_DM	PATH_ID_DM		
OPT_DRIVER_MODEL	OPT_DM	Driver Model: Steering	
VLOW_DRIVER	VLOW_DM	Driver Model: Steering	
TC_THR_APP	TC_TH_APP	Powertrain	Powertrain
TC_THR_REL	TC_TH_RELEASE		

## Known Issues

1. Moving objects can be set to follow a path using a parameter `PATH_ID_OBJ` that identifies the ID number of a path, with an offset calculated by the `LTARG` function using a dataset identified with the parameter `LTARG_ID_OBJ`. If these parameters are changed during an Event, the links to new datasets for the path and/or `LTARG` function were not applied unless the Event triggered a new initialization.

A workaround is to ensure the Event that changes the path ID also reset a parameter identified with an [I] in the Echo file, e.g.,

```
OPT_DIRECTION = OPT_DIRECTION
```

2. Using VS Commands to call any table function with interpolation type STEP returns zero at all times. There is a workaround:
  - Select the interpolation as Linear.
  - Specify the table data to step from one value to another in the shortest possible interval. For example, (X, Y) = (0 1 1.001 2, 1 1 2 2)
3. Road friction is set to 0 for some special cases, such as emulating a K&C test setup. In some cases, the solver crashed. A workaround is to set the friction to a very small value such as 0.001.

## TruckSim 2018.0 vs TruckSim 2017.1

### Bug Fixes and Errata

1. The occlusion of moving objects that are detected by ADAS sensors was based on sorting the objects based on distance from the sensor. The simple method used in past version did not always work as intended in some cases where targets have drastically different sizes.
2. Tire forces and moments at the center of tire contact (CTC) may be imported. However, if the selected tire model was an external model such as TASS MF-Tyre or COSIN/Ftire, the import variables were not used in the multibody model, but were added to the force and moment output variables. This was misleading. In version 2018, these imports are ignored completely for external tire models ( $\text{OPT\_TIRE\_MODEL} \geq 8$ ) and do not produce confusing outputs.
3. Screens for tabular tire data have an option for **Legacy Tire (2D, absolute slip, from zero)**. When selected, the tabular data should use only positive values of slip. However, if asymmetric data were entered that started with negative slip angles, there was no error reporting. This has been fixed. As a result, there may be cases where datasets from previous versions will now generate errors.
4. 5-axle vehicles with 1- or 2-axle driven powertrains had incorrect drive torque calculation due to assigning wrong axles' wheel speeds. This has been fixed.
5. Powertrain may be selected on the lead unit which has more than five axles. However, the powertrain model didn't provide proper calculations for a vehicle unit with six or more axles. In version 2018.0, the powertrain systems are applied on the first five axles if the vehicle unit has more than five axles.
6. Texture render issues which occurred when viewing Mcity scenes in VS Visualizer's compatible mode have been corrected.
7. The `api-ms-win-crt-runtime*.dll` error when using Windows 7 SP1 has been addressed with v2018.0.

8. Run and CreateAllParfile COM functions failure on locked screen has been fixed. Users may have to unlock a screen before executing COM functions which would modify screen data.
9. Compatibility problems with MATLAB 2017a and 2017b have been corrected

## Backward Compatibility

### *Output variables for optional parts*

The default behavior for VS Commands that create new output variables was changed in v2018.0 to define the variables, but not activate them for writing to file. Instead, they are activated as needed for use in animations and plots (the same as built-in output variables).

We have found a few cases where output variables for optional parts (e.g., reference points) are referenced in old databases for use in VS Visualizer for plotting or animation before the variables are created. Because the variables were always activated for writing, plots and animations worked as expected. However, with the new behavior, the variables are not activated because they do not exist when the relevant Parsfiles are read.

A quick workaround is to activate all output variables for writing to file using the checkbox on the **Run Control** screen. However, this can generate unnecessarily large files, especially when the simulation involves many optional parts such as moving objects and ADAS sensors. The recommended fix (used in all 2018 examples) is to make sure links to datasets that define the new parts (reference points, motion sensors, etc.) are processed before links to datasets that use the new output variables in plots or custom animations.

### *Miscellaneous Compatibility Issues*

1. Simfiles must have a vehicle code in order for the VS Solver to activate the appropriate modules. This is done automatically when runs are made from the TruckSim GUI. If running simulations from external software, be sure the Simfile includes the `VEHICLE_CODE` keyword and code.
2. New dynamic state variables were added for hitch deflections (X-Y-Z) and suspensions translations (X-Y). Deflections are typically about 0.5 mm. Although agreement between 2018.0 models and 2017.1 is close (typically the first three or four significant digits), it is not exact. For more information about these effects please see the memo *Modular Vehicle Models in Version 2018*, available from the Help menu: **Help>Release Notes> TruckSim 2018: Modular Vehicle Models**.
3. More error handling has been added in recent versions. When using datasets imported from older versions, especially those prior to version 9, errors might be reported. In most cases, the errors occurred in the older versions but were not noticed.
4. Variables that were listed as state variables but are not truly independent were removed from the lists of state variables. For example, road steer variables (SV\_STR\_L1, SV\_STR\_L2, etc.) are no longer listed. References to these variables are no longer valid. For example, the steering variables were specified for the linearization examples in 2017.1; they were removed from the linearization examples provided in v2018.

5. Output variables used to support fake shadows for animations in the obsolete SurfAnim program have been removed from the solvers.
6. One of the output variables installed by the VS Command `INSTALL_DRIVER_PREVIEW_SENSORS` was renamed for consistency with other outputs. The old name was `X_RdS_i`; the new name is `X_Targi`. Both names are installed and may be used to activate outputs for writing/export. Both may also be used in VS Command formulas. However, only the new name can be used to define plots for VS Visualizer.

<b>Note</b>	The old name <code>Y_RdS_i</code> was intended to be renamed as <code>Y_Targi</code> . However, there was a mistake that was not noticed until the last minute. Therefore, we have a new name for the X coordinate but an old name for the Y coordinate.
-------------	--

7. Examples provided in past versions for running the VS Math Model from custom "wrapper programs" used the old convention of having a separate solver for each vehicle configuration (e.g., `s_s.dll`, `s_ss.dll`, etc.) located in the folder `Programs\solvers\Default` for 32-bit versions and `Programs\solvers\Default64` for 64-bit versions. Those old wrapper programs will not work with the new convention of having just two modular solvers: `Programs\solvers\trucksim_32.dll` and `Programs\solvers\trucksim_64.dll`. Any custom wrapper code that used the old name and locations of the VS Solver DLL will need to be modified, as is done with the example source code provided in v2018.
8. The default behavior for VS Commands that create new output variables was changed in v2018 to define the variables, but not activate them for writing to file. (This is the same behavior as all built-in output variables.) Variables are activated for writing by using them in datasets from the **Plot Setup**, **Animator Reference Frame**, and **I/O Channel Output** libraries.

Be aware that output variables cannot be activated until after they are defined. Therefore, datasets to create new parts (sensors, moving objects, etc.) must be referenced before the datasets that will make use of them.

9. The solver now checks for exceptional floating-point values such as NaN (not a number) at simulation time. When importing a database from an earlier version of TruckSim, some existing VS commands equations may need to be rewritten. For example, the VS Commands equations for engine sounds in the 2017.1 database should be changed to use the `ABS` function before applying the `SQRT` function, to avoid producing a NaN value.

## Known Issues

1. When working in Windows 10 with multiple monitors that have different DPI settings, the menus of TruckSim 2018 do not position correctly on at least one of the monitors. Once a TruckSim menu is activated, the mouse position is offset or wrong until the menu is

dismissed or an item is selected. To avoid this behavior, ensure that the TruckSim pop-up windows are positioned on the screen where you intend to work.

2. When using dSPACE 1006 and 1005 HIL boards, output names greater than 8 characters will not be received by VS Visualizer during live animation. Be sure to use short names on these platforms.

## TruckSim 2017.1 vs TruckSim 2017.0

### Bug Fixes and Errata

The following bugs have been fixed.

#### *Restoring States*

The VS Math Model can routinely save and restore the state of the math model. This capability is used to numerically integrate the ordinary differential equations (ODEs) into the math model, for supporting the VS Commands `SAVE_STATE` and `RESTORE_STATE`, as well as the VS Command `LINEARIZE`.

It was recently discovered that the save/restore operation did not properly handle some state variables which are unrelated to multibody variables. These include brake pressures, some controller integral control variables, and tire lags. In most cases, the effect was not noticeable. However, there are some situations where different results will be obtained with 2017.1 as compared to 2017.0.

1. The VS Commands `SAVE_STATE` and `RESTORE_STATE` did not properly handle cases in which the state being saved or restored to involved significant braking or powertrain acceleration.
2. Most of the numerical integration methods calculate state variables at the half-time step ( $TSTEP/2$ ) to calculate derivatives at that time. The state of the model is then restored to the beginning of the time step to apply the numerical integration methods. When comparing runs made with TruckSim 2017.1 to TruckSim versions 2017.0 and earlier, agreement is usually good for 5 or 6 significant digits. Overall vehicle motions will match when overlaying animations; sometimes, however, a “flicker” is observable if the colors of the vehicle are different between the two runs. In these cases, the results from TruckSim 2017.1 are correct.

**Note** If the numerical integration method is set to Euler (`OPT_INT_METHOD = -1` and a smaller time step is specified), then perfect agreement is obtained between TruckSim 2017.0 and 2017.1.

#### *Configurable Function option: 2D Spline interpolation and extrapolation*

The Configurable Function option 2D Spline interpolation and extrapolation was updated to consider the row and column data simultaneously when creating the spline field. Although the

appearance of the data in the GUI window may appear different in 2017.1 vs. 2017.0, the GUI plot now matches the simulation results.

### *Licensing*

1. Live animation in Simulink did not work unless there was an Extra Live Animator license, which was not supposed to be a requirement.
2. Some real-time platform add-on licenses (Frame Twist, Sensors) denied usage rights even when the license feature was properly enabled.
3. Licenses on some real-time platforms excluded the possibility to make multiple runs.
4. Licenses could be lost after 10 minutes when using mixed license sources (e.g. network and dongle license).

### *Miscellaneous*

1. When a moving object is set to follow a user-specified path that is also the reference path for the road surface on which the moving object is traveling, there should be no ambiguity in locating the object and obtaining its correct Pitch angle, Roll angle, and Z coordinate. In TruckSim versions 2017.0 and earlier, a generalized method was used in which the incorrect location might be used in cases of looped roads.
2. In some cases, using clothoids to connect positive to negative curvature could cause the clothoid calculation to inject a discontinuity into the path.
3. Plot bounds were calculated incorrectly by VS Visualizer for interactively created plots.
4. VS Visualizer could crash when loading OSGB files (i.e., binary .osg files, such as those used for the Mcity examples). Warning messages have been added when OSGB files are included in the creation of a VSRAP file.
5. Clicking the **Plot** button on the **Batch Run** screen with the output set to VS format failed to produce a visible plot.
6. The user-specified ROAD\_DZ\_ID value was not retrievable when a new run was made using the END file from an existing run to continue the simulation.

The VS Command function SRAND produced differing sets of pseudo-random numbers if the float value passed to SRAND was outside the range of -2,147,483,648 and 2,147,483,647. For TruckSim 2017.1, the SRAND command now casts the floating point parameter to an unsigned integer rather than a signed integer.

### **Known Issues**

1. When working with Windows 10 and multiple monitors, and the monitors have different DPI settings, the menus of TruckSim 2017 do not position correctly on at least one of the monitors. Once a TruckSim menu is activated, the mouse position is offset/wrong until the menu is dismissed or an item is selected. To avoid this behavior, ensure that the TruckSim pop-up windows are positioned on the screen where you intend to work.

2. The internal algorithm for handling occlusion of moving objects in ADAS Sensor detections is based on several detection points (left edge, closest, right edge) and an implicit assumption that objects are similar in size. When one object is significantly smaller than others (0.2-m target used for lane detection vs. a 5-m vehicle), then the smaller objects are sometimes not occluded as expected.

## TruckSim 2017.0 vs TruckSim 2016.2

### Bug Fixes and Errata

The following bugs have been fixed.

1. In certain transient situations, the closed-loop speed controller could set the master cylinder pressure negative, attempting to generate positive wheel torque. The brake moment always resists the rolling direction, so the effect was that the controller generated brake torque. Although the physics remained valid, this bug sometimes affected the controlled behavior in an unintended manner.
2. References to road and path ID numbers in algebraic expressions could crash TruckSim if the ID was specified with a number, e.g., the function call `ROAD_SSTART_ID(1212)`. The problem was that the ID (e.g., 1212) would not be valid until the model initialized, which is after the files are read. The error handling was changed for 2017, and some automatic preprocessing was added to allow the use of ID numbers in algebraic expressions.
3. The hitch model includes options for friction in the three hitch moments. The friction support variables were not updated properly, causing distortion in the friction part of the moment.
4. In versions prior to 9.0 (2015), tables made with the VS Command `DEFINE_TABLE` supported the option to use an algebraic formula to calculate the result from the Configurable Function. This capability was accidentally removed in versions 9 and 2016; it is now restored.
5. When off-center payloads (those with non-zero Y coordinates) were added to the model, the XY and YZ products of inertia due to the location of the masses were calculated incorrectly.
6. The speed parameter passed to the Steering Parking Torque configurable functions was the signed value of speed. This could produce unintended results for a vehicle traveling in reverse (negative speed). The function now receives the absolute value of speed.
7. Setting tire relaxation length to zero could cause a divide by zero error at initialization.
8. A parameter `VLOW_DAMP_Y (iaxle, iside, itire)` for advanced use (not supported by the SGUI, set only by typing into yellow fields) had a default value of 0.5 km/h. This parameter causes tire lateral force damping to be increased below the specified speed. It can be valuable in certain applications such as driving simulators. However, it can also contribute to instability in certain situations, notably in vehicles with large caster angles. The default

value for this parameter has been changed to zero, but the parameter itself has been retained for advanced users to set by typing into a miscellaneous field.

## Backward Compatibility

### *Occlusion Calculation for ADAS Sensors and Sensor Targets*

Updates have been made to the calculation of occlusion for sensor targets when used with the ADAS Range and Tracking Sensors. The internal algorithm for handling occlusion involves sorting the objects based on proximity. In TruckSim 2016.2 and earlier, the sorting was based on the distance to the closest point for each object. An effect is that sometimes a large object such as a wall would be used incorrectly to occluding smaller objects.

The sorting order is changed for TruckSim 2017 to use the furthest of the points found by checking both the left and right edges. This solves the original problem of having a false occlusion. However, a new effect (less common) is that sometimes, small objects that should be occluded are not. For example, in the ADAS examples that use sensor targets for traffic vehicles and lane edge markers, the ADAS sensor may detect the lane marker if it is slightly behind the corner of a traffic vehicle. This is noted later in the Known Issues section.

### *Initialization using a Path*

When the vehicle is initialized relative to a path, there were some ambiguities in past versions if the driver model was not enabled. In 2017, the parameter `PATH_ID_DM` always identifies the path used to set initial position and heading. `PATH_ID_DM` is always set by default to the user ID of the most recently defined path. If another path should be used, then `PATH_ID_DM` must be set explicitly after all path datasets are loaded. There may be a few datasets in past versions where this was not true.

### *Paths and Road Surfaces*

1. The user ID numbers for paths and road surfaces (`PATH_ID` and `ROAD_ID`, respectively) are restricted in version 2017 to either match the internal ID (e.g., `PATH_ID(3)=3`) or they must be assigned values 999 or greater. This is done to avoid conflicts when the maximum number of paths and roads is increased in future versions beyond the current limit of 100. Past versions did not restrict the values that could be set to `PATH_ID` and `ROAD_ID`. If datasets from previous versions used ID numbers that did not match the internal ID numbers and were less than 999, then they must be renumbered to work in version 2017.
2. In past versions, the road surface model has included two incremental DZ layers for each road, provided with the Configurable Function `ROAD_DZ`. The 2017 version extends the model to include any number of layers ranging from 0 to 100. Links in old datasets to a flat `ROAD_DZ` dataset are unnecessary. The new examples from Mechanical Simulation do not link to a layer dataset unless it adds 3D information. However, links in old databases are not automatically removed. They do not result in any harm, but they do add some complexity that is no longer necessary.



3. The indexing used to access a specified ROAD\_DZ dataset was changed from two indices (road ID and 1 or 2) to one (IROAD\_DZ). A user ID was added to the ROAD\_DZ Configurable Function. When road surfaces are assembled from the GUI, the changes are handled automatically. However, advanced users who specify road information without going through the GUI might have to change some files.
4. Versions prior to 2001 used a rectangular grid to provide a 3D surface. Two legacy library screens existed for specifying elevation and friction using tables of X-Y coordinates. In version 2016, the grid option was removed, as it is the same as a straight road ( $S = X$ ,  $L = Y$ ). These X-Y grid screens had red text announcing they would be removed in version 2017, and they are now gone.

### *Import Variable Names*

User-defined import variables should be named with a prefix `IMP_`, e.g., `IMP_CTRL_1`. If the prefix is not included (e.g., the import is named `CTRL_1`), then TruckSim cannot properly scan the data when connecting to external models via Simulink, FMU, and other methods.

Past versions had a checkbox on the **Preferences** screen labeled “Scan for `IMP_` and `EXP_` commands and send last to math model.” If unchecked, the scanning was not performed, and newer connection methods could not be used. This checkbox was removed from the 2017 version, and an error message is generated if an import variable is defined that does not begin with `IMP_`.

The recommended syntax is:

```
define_import Imp_NewVar  
  
import Imp_NewVar Add 0
```

where the number zero (0) represents the user-defined initial value of the import variable at the beginning of the simulation. Depending on the intended use of the new import variable, this initial value can be any number; supported options include integers (0, 1, 2, etc.), fractions (2/3), and symbols recognized by the GUI such as `pi`. If the new import variable is activated correctly, it is written at the end of the Echo File.

### *Retired Variables*

Older versions of TruckSim included variables for supporting animation of dashboard dial gauges. VS Visualizer provides native support for HUD displays making the old method obsolete. The old variables were named `A_Comp`, `A_Speedo`, and `A_Tach`. If needed, they can be redefined with VS Commands.

### *Retired calculated parameters*

Older versions of TruckSim included numerous vehicle-level parameters that were calculated from the math model parameters. These include static axle loads, total masses and inertias, and CG locations of various combinations of masses in the model. Figure 3 shows the top of the section, with static loads. Figure 4 shows the rest of the section, with calculated properties that are based on combinations of masses in the model.

The screenshot shows a ConTEXT application window with the title bar "ConTEXT - [C:\Product\_Working\2016.2 Testing\TruckSim2016.1\_Data\Runs\Run\_3d47e539-14...". The menu bar includes "File", "Edit", "View", "Project", "Tools", "Options", "Window", and "Help". The title bar of the active window is "Run\_3d47e539-147a-4376-b7cf-f1d5d7270a81\_ECHO.PAR #". The main text area displays the following content:

```

4350 ! -----
4351 ! CALCULATED VEHICLE MASS AND LOAD PROPERTIES
4352 ! -----
4353 ! The following inertia properties and axle loads with the keyword CALC in the name
4354 ! are approximate, and are provided as a quick check to see that that the vehicle
4355 ! mass and load parameters provide overall vehicle properties that match
4356 ! expectations. They are calculated for the vehicle in its design load condition,
4357 ! with zero pitch and roll angles. Instant loads and coordinate locations from the
4358 ! dynamic model are available as output variables.
4359
4360 ! Calculated static axle loads FZA_L and MZA_L: laden vehicle
4361 ! FZA_L(1) 36229.78386 ; N ! CALC -- Static load, axle 1, laden vehicle [I]
4362 ! FZA_L(2) 68603.30464 ; N ! CALC -- Static load, axle 2, laden vehicle [I]
4363 ! MZA_L(1) 3694.4098 ; kg ! CALC -- Static load, axle 1, laden vehicle [I]
4364 ! MZA_L(2) 6995.5902 ; kg ! CALC -- Static load, axle 2, laden vehicle [I]
4365
4366 ! Calculated static axle loads FZA_UL and MZA_UL: unladen vehicle
4367 ! FZA_UL(1) 24828.73419 ; N ! CALC -- Static load, axle 1, unladen vehicle [I]
4368 ! FZA_UL(2) 50584.40431 ; N ! CALC -- Static load, axle 2, unladen vehicle [I]
4369 ! MZA_UL(1) 2531.826281 ; kg ! CALC -- Static load, axle 1, unladen vehicle [I]
4370 ! MZA_UL(2) 5158.173719 ; kg ! CALC -- Static load, axle 2, unladen vehicle [I]
4371
4372 ! Wheel-center heights H_WC_UL (axle average) for the vehicle unladen.
4373 ! H_WC_UL(1) 507.3322785 ; mm ! CALC -- Approximate wheel-center height for axle 1
4374 ! [I]
4375 ! H_WC_UL(2) 507.0958152 ; mm ! CALC -- Approximate wheel-center height for axle 2
4376 ! [I]
4377

```

The status bar at the bottom shows "Ln 1, Col 1", "Insert", "Sel: Normal", "DOS", and "File size: 227085".

Figure 3. Echo file section showing calculated mass and load parameters in 2016.1.

**Note** This subsection did not appear in the original backward compatibility document for 2017.0; it was added July 2019.

```

CONTEXT - [C:\Product_Working\2016.2 Testing\TruckSim2016.1_Data\Runs\Run_3d47e539-14...
File Edit View Project Tools Options Window Help
Run_3d47e539-147a-4376-b7cf-f1d5d7270a81_ECHO.PAR #

4378 ! Calculated properties for TPL: total payload for vehicle
4379 ! LX.CG_TPL 2750 ; mm ! CALC -- X distance TPL CG is behind vehicle origin [I]
4380 ! Y.CG_TPL 0 ; mm ! CALC -- Y coord. of TPL CG [I]
4381 ! H.CG_TPL 1750 ; mm ! CALC -- Height (Z coord.) of TPL CG [I]
4382 ! M_TPL 3000 ; kg ! CALC -- Mass of TPL [I]
4383 ! W_TPL 29419.95 ; N ! CALC -- Weight of TPL [I]
4384 ! IXX_TPL 2187.5 ; kg-m2 ! CALC -- Roll moment of inertia of TPL [I]
4385 ! IYY_TPL 12500 ; kg-m2 ! CALC -- Pitch moment of inertia of TPL [I]
4386 ! IZZ_TPL 14187.5 ; kg-m2 ! CALC -- Yaw moment of inertia of TPL [I]
4387 ! IXY_TPL 0 ; kg-m2 ! CALC -- XY product of inertia of TPL [I]
4388 ! IXZ_TPL 0 ; kg-m2 ! CALC -- XZ product of inertia of TPL [I]
4389 ! IYZ_TPL 0 ; kg-m2 ! CALC -- YZ product of inertia of TPL [I]
4390
4391 ! Calculated properties for SL: vehicle sprung mass when laden
4392 ! LX.CG_SL 2991.217949 ; mm ! CALC -- X distance SL CG is behind vehicle origin [I]
4393 ! Y.CG_SL 0 ; mm ! CALC -- Y coord. of SL CG [I]
4394 ! H.CG_SL 1376.282051 ; mm ! CALC -- Height (Z coord.) of SL CG [I]
4395 ! M_SL 9360 ; kg ! CALC -- Mass of SL [I]
4396 ! W_SL 91790.244 ; N ! CALC -- Weight of SL [I]
4397 ! IXX_SL 10499.73462 ; kg-m2 ! CALC -- Roll moment of inertia of SL [I]
4398 ! IYY_SL 44155.93173 ; kg-m2 ! CALC -- Pitch moment of inertia of SL [I]
4399 ! IZZ_SL 45226.79712 ; kg-m2 ! CALC -- Yaw moment of inertia of SL [I]
4400 ! IXY_SL 0 ; kg-m2 ! CALC -- XY product of inertia of SL [I]
4401 ! IXZ_SL -398.0096154 ; kg-m2 ! CALC -- XZ product of inertia of SL [I]
4402 ! IYZ_SL 0 ; kg-m2 ! CALC -- YZ product of inertia of SL [I]
4403
4404 ! Calculated properties for TL: total vehicle when laden
4405 ! LX.CG_TL 2938.278765 ; mm ! CALC -- X distance TL CG is behind vehicle origin [I]
4406 ! Y.CG_TL 0 ; mm ! CALC -- Y coord. of TL CG [I]
4407 ! H.CG_TL 1269.925164 ; mm ! CALC -- Height (Z coord.) of TL CG [I]
4408 ! M_TL 10690 ; kg ! CALC -- Mass of TL [I]
4409 ! W_TL 104833.0885 ; N ! CALC -- Weight of TL [I]
4410 ! IXX_TL 11980.87194 ; kg-m2 ! CALC -- Roll moment of inertia of TL [I]
4411 ! IYY_TL 51928.34327 ; kg-m2 ! CALC -- Pitch moment of inertia of TL [I]
4412 ! IZZ_TL 52634.07133 ; kg-m2 ! CALC -- Yaw moment of inertia of TL [I]
4413 ! IXY_TL 0 ; kg-m2 ! CALC -- XY product of inertia of TL [I]
4414 ! IXZ_TL 54.82862301 ; kg-m2 ! CALC -- XZ product of inertia of TL [I]
4415 ! IYZ_TL 0 ; kg-m2 ! CALC -- YZ product of inertia of TL [I]
4416
4417 ! Calculated properties for TU: total vehicle when unladen
4418 ! LX.CG_TU 3011.729519 ; mm ! CALC -- X distance TU CG is behind vehicle origin [I]
4419 ! Y.CG_TU 0 ; mm ! CALC -- Y coord. of TU CG [I]
4420 ! H.CG_TU 1082.639792 ; mm ! CALC -- Height (Z coord.) of TU CG [I]
4421 ! M_TU 7690 ; kg ! CALC -- Mass of TU [I]
4422 ! W_TU 75413.1385 ; N ! CALC -- Weight of TU [I]
4423 ! IXX_TU 8832.223412 ; kg-m2 ! CALC -- Roll moment of inertia of TU [I]
4424 ! IYY_TU 38319.36041 ; kg-m2 ! CALC -- Pitch moment of inertia of TU [I]
4425 ! IZZ_TU 38298.737 ; kg-m2 ! CALC -- Yaw moment of inertia of TU [I]
4426 ! IXY_TU 0 ; kg-m2 ! CALC -- XY product of inertia of TU [I]
4427 ! IXZ_TU 431.7778908 ; kg-m2 ! CALC -- XZ product of inertia of TU [I]
4428 ! IYZ_TU 0 ; kg-m2 ! CALC -- YZ product of inertia of TU [I]
4429

Ln 1, Col 1 Insert Sel: Normal DOS File size: 227085

```

Figure 4. Calculated mass and inertia properties with payloads.

This section of the Echo file was removed in 2017.0. Some of the parameters were moved to other sections of the Echo file. For example, static axle loads were moved to the Suspension sections; total payload properties were moved to the Payloads section. Others were discontinued. Table 17 lists all of the parameters that were discontinued in the 2017.0 release. Note that the inertia properties for the loaded sprung mass were removed in 2017.0 and then restored in 2018.0, calculated more rigorously with the new modular architecture.

Table 17. Calculated parameters that were discontinued in 2017.0.

Parameter (root)	Component	Description	Status
MZA_L, MZA_UL	Axle	Static load, kg	Discontinued
H_WC_UL	Axle	Static wheel height	
LX_CG, Y_CG, H_CG, M, W, IXX, IYY, IZZ, IXY, IXZ, IZZ	Total payload: suffix _TPL	CG location, mass, weight, moments and products of inertia	
LX_CG, Y_CG, H_CG, M, W, IXX, IYY, IZZ, IXY, IXZ, IZZ	Loaded sprung mass: suffix _SL	CG location, mass, weight, moments and products of inertia	Restored in 2018.0 (except W)
IXX, IYY, IZZ	Loaded vehicle unit: suffix _TL	Instant moments of inertia	Restored in 2019.0
W, IXY, IXZ, IZZ		Weight, products of inertia	Discontinued
LX_CG, Y_CG, H_CG, M, W, IXX, IYY, IZZ, IXY, IXZ, IZZ	Unloaded vehicle unit: suffix _TU	Instant CG location, mass, weight, moments and products of inertia	

**Note** To save space in the table, root parameters are shown, e.g., M. The extension associated with the component is listed separately, e.g., \_SL. The parameter listed in Figure 4 for the mass of the sprung mass when laden is the combination of the root name and extension: M\_SL.

### Support for SurfAnim Animator and WinEP Plotter

The legacy animator SurfAnim and plotter WinEP are no longer supported, to provide full support for VS Visualizer. For example, the use of translucent 2D shapes to resemble shadows in SurfAnim is obsolete given the true shadows shown in VS Visualizer. GUI controls for workarounds needed for the legacy tools were identified in past versions with red text, announcing they would be removed in 2017; they are removed in 2017.

WinEP supported a moving average filter option, with a **Plot: Transform** screen for settings. It has not been used in any examples for over ten years and is not supported in VS Visualizer. That screen has been removed.

### OPT\_DRIVER\_EXT Removal

As part of the process for modularizing the solver the parameter OPT\_DRIVER\_EXT was removed. This parameter was originally introduced to allow users to introduce faults into a power steering system without interference from the closed loop steering controller. However, it never functioned as intended as the steering controller was able to “detect” the additional steering input indirectly through the additional error to the path. The introduction of torque control as an option for the closed loop speed controller in the 2019.1 release instead allows fault injection to be handled much as it would in the physical test.

## Known Issues

1. The internal algorithm for handling occlusion of moving objects in ADAS Sensor detections is based on several detection points (left edge, closest, right edge) and an implicit assumption that objects are similar in size. When one object is significantly smaller than others (0.2-m target used for lane detection vs a 5-m vehicle), then the smaller objects are sometimes not occluded as expected.
2. Eight variables that are sometimes used to locate a virtual video camera were removed from TruckSim 2016.2 to avoid situations where they were not applicable and calculations might fail. The variables can be installed with a new command `INSTALL_CAMERA_OUTPUTS`. Databases for previous versions do not include this command. Hence, the command must be added to older datasets in order to use the road-based camera tracking option.
3. If the extended table-lookup tire model is used to include nonlinear inclination effects, the influence of friction is not applied to inclination effects.
4. Tire forces due to camber are not influenced by friction in the same way as tire forces due to slip. The issue is that camber-induced forces do not saturate in the absence of slip in the current model. (The similarity method used for handling friction uses combined lateral and longitudinal slip, but does not include large inclination effects.) The sensitivity to friction is still under investigation.
5. The Road: X-Y-Z Coordinates of Edges screen can crash when only two rows of data are used for each table. The requirement is that three or more rows of data must be used for each table when entering the edge coordinates. Work is on-going to improve this screen.
6. When working in Windows 10: If more than one monitor is being used, and the monitors have differing DPI settings, then the menus of TruckSim 2017 do not position correctly on one of the monitors. Once the menu is activated, the mouse position is offset/wrong until the menu is dismissed or an item is selected. Mouse coordinates are scaled by the DPI setting of the other monitor. To avoid this issue, ensure that the DS product / pop-up windows are positioned on the screen you intend to work on.

## TruckSim 2016.2 vs TruckSim 2016.1

### Bug Fixes

The following bugs were identified and fixed in the VS Math Model and VS Browser.

#### *Bug Fixes in the VS Math Model*

1. The VS Math Model added memory each time step, causing problems if long runs were made on systems with limited memory resources such as many RT systems. This bug was introduced in version 2016.1.
2. Setting units for a parameter in the same line as the value (e.g., `LX_AXLE(2) = 100; in)` did not work. This bug was introduced in version 2016.0.

3. The VS Command `ROAD_Z_ID` and corresponding API function `vs_road_z_id` returned station rather than Z. This bug was introduced in version 9.0.
4. In certain situations, the closed-loop speed controller set a command brake pressure together with a throttle request. This behavior is rare, but had existed for years.
5. A minor quantization effect could occur when path calculations were made outside the range of an X-Y table or parametrized path segments. This bug was introduced in version 9.0.
6. The built-in Pac52 tire model would not initialize properly if the simulation started at zero speed. Other models could fail at zero speed if the relaxation length were set to zero. This behavior had existed for years.
7. The option to set an equation for a user-defined Configurable Function did not work (e.g., for a user-defined function `MY_FUNC`, the keyword `MY_FUNC_EQUATION` was not installed). This bug was introduced in version 9.0.
8. Some error handling situations have been improved, preventing software crashes in response to bad data.

### *Bug Fixes in the VS Browser*

1. COM commands to change blue links failed in some conditions where the link type was fixed to a particular library.
2. Changing the number of rows in a dataset from the **Path/Road Segment Builder (Legacy)** library corrupted other dataset in the library. The only workaround was to quit TruckSim and rebuild the database. This bug was introduced in 2016.0.
3. The number of rows in the **Generic Table** screen was sometimes set wrong.
4. The clothoid option is not valid for the first or last segment the **Path Segment Builder** screen. Instead of giving error message, the VS Browser crashed if a bad setting was attempted
5. The **Path/Road Segment Builder (Legacy)** screen showed an incorrect number of roads for a new dataset.
6. If a dataset in the **Path/Road Segment Builder (Legacy)** library had starting X-Y coordinates and path heading set to zero, the values were not written to the dataset Parsfile. This had no effect on the simulation results because the default values for those parameters are zero. However, it could be confusing because the yellow fields were sometimes blank. This bug was introduced in 2016.1.
7. If the option to select an un-steered suspension type was selected on a **Vehicle Lead Unit** screen, the text for the selection did not fit in the drop-down control. This was a cosmetic bug, introduced in 2016.0.

### **Backward Compatibility**

Most of the bug fixes apply to uncommon situations that do not appear in the example datasets provided with TruckSim.

One exception is that the bug in the closed-loop speed controller can cause minor differences in simulations where the speed controller is given target speeds that involve significant acceleration. The old controller could apply brief braking inputs incorrectly, leading to small differences in vehicle speed.

As noted in the release notes for TruckSim 2016.1, eight variables that are sometimes used to locate a virtual video camera were removed to avoid situations where they were not applicable and calculations might fail. The variables can be installed with a new command `INSTALL_CAMERA_OUTPUTS`. Databases for previous versions do not include this command. Hence, the command must be added to older datasets in order to use the road-based camera tracking option.

## Known Issues

1. The internal algorithm for handling occlusion of moving objects in ADAS Sensor detections is based on several detection points (left edge, closest, right edge) and an implicit assumption that objects are similar in size. When one object is significantly larger than others (e.g., a 50-m wall behind 5-m vehicles), then the smaller objects might be occluded incorrectly. The workaround is to replace the large object with multiple connected smaller objects.
2. Datasets from the **Preferences** library include the status of a checkbox labeled “Scan for IMP\_ and EXP\_ commands and send last to math model.” The TruckSim example datasets installed in recent versions always have this box checked.
3. This box must be checked when using the new Generation-2 TruckSim S-Function, or when running any TruckSim RT setup.
4. The behavior when the box is not checked supports old datasets where VS Commands were used to define new import variables without an “IMP\_” prefix (not recommended), and they were activated without using the IMPORT command (not recommended).
5. This checkbox will be removed in future versions.
6. If the extended table-lookup tire model is used to include nonlinear inclination effects, the influence of friction is not applied to inclination effects.
7. Tire forces due to camber are not influenced by friction in the same way as tire forces due to slip. The issue is that camber-induced forces do not saturate in the absence of slip in the current model. (The similarity method used for handling friction uses combined lateral and longitudinal slip, but does not include large inclination effects.) The sensitivity to friction is still under investigation.

## TruckSim 2016.1 vs TruckSim 2016.0

### Bug Fixes and Errata

The following bugs have been fixed.

1. When the TruckSim model was initialized using station along the driver model path, the station was sometimes incorrectly reset to a value that is also valid for the initial X-Y

- coordinates of the vehicle, but in the vicinity of  $S=0$ . If this happened, the driver model did not perform correctly. Other times, the reset failed to find a value of  $S$  and the simulation did not run. This problem only occurred when starting the simulation at a location far from the origin.
2. The **Batch Matrix** screen has a button **Create Matrix and Make All Runs** that causes a set of **Run Control** datasets to be created and used to immediately run a simulation. The new datasets are based on an example **Run Control** dataset used as a template. If the template example did not have a linked vehicle dataset, TruckSim failed to load the VS Solver DLL.
  3. When importing data from version 9 the road shapes would sometimes be missing when the run was animated if the imported dataset was never refreshed.
  4. The **Update Road Surface 3D Shape Files** button on the **Road: 3D Surface (All Properties)** screen could cause a crash if invalid data were linked to the road. TruckSim now properly handles the error.
  5. In some simulations of K&C tests, it is convenient to remove tire forces by setting the tire/ground friction coefficient to a very small value (e.g.,  $\mu < 0.01$ ). Internal table-lookup tire models had a bug such that small longitudinal and lateral forces were generated if the tire longitudinal slip were positive.
  6. A parameter `L_SPG_ADJ` provides wedge adjustment for the upper spring height. The effect was double-counted in the code, resulting in an adjustment with twice the specified value. (This feature is rarely used.)
  7. The dSPACE 1006 RT license checking for the frame-twist option did not work.
  8. The audio sound set for engine noise were moved in TruckSim 2016 from animator datasets to the engine dataset, in support for advanced users who might add new sound sets for different vehicles. The dataset labeled **General Vehicle Sounds** was renamed to **Wind Noise**. These minor changes have no effect on existing example datasets.
  9. The **Road: Off-Center Elevation, Variable Width** screen would erroneously display a locked dataset error when the user manually locked a dataset.

## Known Issues

1. The internal algorithm for handling occlusion of moving objects in ADAS Sensor detections is based on several detection points (left edge, closest, right edge) and an implicit assumption that objects are similar in size. When one object is significantly larger than others (e.g., a 50-m wall behind 5-m vehicles), then the smaller objects might be occluded incorrectly. The workaround is to replace the large object with multiple connected smaller objects.
2. The Closed Loop Speed Controller includes an option to enable engine braking as part of the speed control. In certain situations, this can result in the Master Cylinder Pressure momentarily going negative. If the checkbox to include engine braking is not checked, this behavior does not occur.



3. As noted earlier, eight variables that are sometimes used to locate a virtual video camera were removed from TruckSim 2016.1 to avoid situations where they were not applicable and calculations might fail. The variables can be installed with a new command `INSTALL_CAMERA_OUTPUTS`. Databases for previous versions do not include this command. Hence, the command must be added to older datasets to use the road-based camera tracking option.
4. Datasets from the **Preferences** library include the status of a checkbox labeled “Scan for IMP\_ and EXP\_ commands and send last to math model.” The TruckSim example datasets installed in recent versions always have this box checked.
5. This box must be checked when using the new Generation-2 TruckSim S-Function, or when running any TruckSim RT setup.
6. The behavior when the box is not checked supports old datasets where VS Commands were used to define new import variables without an “IMP\_” prefix (not recommended), and they were activated without using the `IMPORT` command (not recommended).
7. This checkbox will be removed in future versions.
8. The error handling in versions 9 through 2016.1 does not handle a certain type of error well; instead of generating an error message, the solver crashes.
9. Values assigned to parameters in a Parsfile that involve symbolic function calls are evaluated instantly if the argument to the function is numerical. For example, `sin(30)` is instantly evaluated to give a numerical value of 0.5 (the sine of 30°). This behavior does not work for road and path functions that provide a road or path property based on a User ID, because the roads and paths are not installed until after the entire set of input Parsfiles has been scanned. For example, consider the commands:

```
DEFINE_PARAMETER NEW_SSTART; m; station start for 1212
EQ_INIT NEW_SSTART = ROAD_SSTART_ID(1212)
```

10. This will not work because the user ID 1212 is not recognized when the Parsfile is being scanned.
11. The workaround is to introduce a parameter and pass the parameter to the path or road function, as in this example:

```
DEFINE_PARAMETER NEW_SSTART; m; station start for 1212
DEFINE_PARAMETER TEMP_UID 1212; -; temp user defined road id
EQ_INIT NEW_SSTART = ROAD_SSTART_ID(TEMP_UID)
```

12. The error handling will be improved in future versions.
13. If the extended table-lookup tire model is used to include nonlinear inclination effects, the influence of friction is not applied to inclination effects.
14. Tire forces due to camber are not influenced by friction in the same way as tire forces due to slip. The issue is that camber-induced forces do not saturate in the absence of slip in the current model. (The similarity method used for handling friction uses combined lateral and

longitudinal slip but does not include large inclination effects.) The sensitivity to friction is still under investigation.

## TruckSim 2016.0 vs TruckSim 9.0.2

### Bug Fixes and Errata

The following bugs were identified and corrected.

1. The VS Browser often flickered or flashed when processing COM commands in minimized mode. This has been cleaned up.
2. A VS Browser startup crash has been fixed when the initial license screen is forced up as a reminder for renewal when processing COM commands.
3. Past versions of TruckSim supported five methods of numerical integration. The Adams-Bashforth 2nd Order Method (AB-2) differs from the others by only doing one calculation of the derivatives of state variables at the designated time step. An error in the code was identified that caused the calculation to use a less accurate Euler integration method. The error was fixed, and an option was added to use the Euler method. Thus, there are now six supported methods of numerical integration.
4. The Road X-Y-Z Coordinates of Edges screen did not always write the Parsfile correctly if the user switched to use the lower table as the reference.
5. The VS Math Model did not correctly maintain heading continuity for X-Y Table segments embedded in multi-segment reference paths. The option to include X-Y tables in a multi-segment path was not supported in the GUI, so this bug was not relevant unless advanced users generated datasets outside the GUI. (The bug did not affect single-segment paths.)
6. Sorting of detections for each sensor did not exclude objects with magnitudes of zero unless they were occluded.
7. The option to specify speed of multiple moving objects with target speed vs. station on the **Traffic Motion** screen (now named the **Multiple Moving Objects** screen) generated a VS Command that incorrectly used time instead of station.
8. Preview Points for External Driver Control. In TruckSim version 9.0, the number of supported points was extended from 5 to 10. However, the output variable `Lx_Sen_10` contained 9 characters, one more than the maximum allowed. For TruckSim 2016, the 10<sup>th</sup> preview point has been removed.
9. TruckSim 9.0 introduced three options in a ring control in the upper right corner of the Road: 3D Surface (All Properties) screen: New road with automatic ID; New road with custom ID; and Replace data for existing road. The third option was intended to allow advanced users to override road properties using Events. In TruckSim 2016, this option has been removed, and the recommended method is to refer to the road ID directly in VS Event datasets.
10. A review of the tire data used for the Internal Table Look-up tire model indicated that some of the Aligning Moment data was too large. New Aligning Moment data has been created

for the following tires: 2000 kg Load Rated Tire; 3000 kg Load Rated Tire, and 3500 kg Load Rated Tire.

## Backward Compatibility

### *Simulations with No Road*

Prior versions of TruckSim allowed a simulation to be run without defining a road surface. Starting with version 2016, all simulations include at least one road surface.

To support existing databases, the old ground libraries are still included, with these changes:

1. The legacy library **Ground Elevation, X-Y Grid** is now named **Road: Elevation Map (Legacy)**.
  - a. Each Parsfile from this library adds a Road to the model with the `DEFINE_ROADS` command, and a straight-line path added with the `DEFINE_PATHS` command. This ensures that the new road has compatible coordinates:  $S = X$  and  $L = Y$ .
  - b. The Parsfile has data for Z written for the Configurable function `ROAD_DZ`, with `IROAD` set to the number of the new road and `IDZ_ROAD = 1`.
2. The legacy library **Ground Friction, X-Y Grid** is now named **Road: Friction Map (Legacy)**. The friction data are written for the Configurable function `MU_ROAD`.

These legacy libraries have not been used for examples shipped with TruckSim for years; they are scheduled to be removed in the TruckSim 2017 release.

### *Static Readme Files*

Prior versions of TruckSim included a set of static text readme files to support browsing for the **Import** and **Export** library screens, and to provide some information about the math model. These files do not contain any information about variables added at runtime. Starting with version 9, dynamic machine-generated text and spreadsheet files are generated as needed from the **Run Control**, **Import**, **Export**, **Write**, and **Plot Setup** screens. The dynamically generated files include all variables of a given type (Import, Output, State Variable), including those added with optional modules and even VS Commands.

Many variables that used to be fixed in the VS Solver are now added at runtime (with many more options), such that the static text readme files are too limited.

Datasets imported from older versions of TruckSim for the **Import**, **Export**, and **Write** libraries might have yellow fields that specify old readme files that no longer exist. This does not affect the dataset but does limit the capability for browsing. In case you want to modify an imported dataset that references an old readme file, you should use the link on the screen to an existing Run Control dataset. When you do this, documentation files are automatically generated that are current and complete.

### *Library Titles and Menu Item Names*

If you are a long-time user of TruckSim, you might notice that some items on the **Libraries** menu have been renamed, as have some of the libraries.

As the number of libraries in TruckSim has increased, some of the names have been changed to provide better consistency. For example, TruckSim includes sensors for motion detection (e.g., accelerometers), sensors for internal controllers (ABS, ESC, etc.), and ADAS sensors (radar, video, etc.). The Libraries menu has a submenu **ADAS Sensors and Moving Objects** that was renamed from **Traffic, Objects, and Sensors** in earlier versions.

All libraries from version 9 still exist in version 2016, and names in old databases are automatically converted to the new names.

### *The Original Calculator Tool*

Prior versions have included two Calculator Library tools: **Calculator: Original** and **Calculator: Symbolic**. The **Calculator: Original** tool has been obsolete since the introduction of the **Calculator: Symbolic** tool. It has been removed from TruckSim 2016.

### *Discontinued Units*

TruckSim includes a **Run Control** example that demonstrates how to use VS Commands to redefine units, which for TruckSim 2016 is named: Data Output Options -> SI Units Data Output. The following units have been discontinued for TruckSim 2016: KJ/KG/C2, KJ/KG/C, MPA/(MM3/S), N-M/DEG/S, N/MM/S, MPA/N, MM3/S, C, MM3, and 1/S

## **Known Issues**

1. The internal algorithm for handling occlusion of moving objects in ADAS Sensor detections is based on several detection points (left edge, closest, right edge) and an implicit assumption that objects are similar in size. When one object is significantly larger than others (e.g., a 50-m wall behind 5-m vehicles), then the smaller objects might be occluded incorrectly. The suggested implementation is to replace the large object with multiple smaller connected objects such that the built-in assumptions remain valid.
2. The Closed Loop Speed Controller includes an option to enable engine braking as part of the speed control. In certain situations, this can result in the Master Cylinder Pressure momentarily going negative. If the checkbox to include engine braking is not checked, this behavior does not occur.