SuspensionSim New Features

This document lists notable new features in SuspensionSim going back to version 2017.0.

> **Note** SuspensionSim's version number is based on the version of CarSim which it is released alongside, causing some skips in major and minor version numbers from this perspective.

SuspensionSim 2024.2 New Features

VS Math Model

Suspension Design Factors (SDFs)

The Suspension Design Factors (SDF) feature has been updated to calculate the screw axis of the suspension knuckles. This is activated with the new parameter `OPT_SCREW_AXIS_CALC`, assuming SDFs have been activated. This calculation is disabled by default. The screw axis outputs give the screw motion which takes the initial pose of the knuckle body to its current pose. The screw axis is represented geometrically by a direction vector, position vector, rotation about the axis, and displacement along the axis. See the SuspensionSim Commands Reference Manual for more information.

VS Browser

Database Builder

The Database Builder tool was revised to identify specialty CPAR archives more easily for examples that are not included in the main groups of examples. The screen now has two options for showing CPAR archive files:

1. Show specialty archives that are not applicable for most users. However, there are currently no specialty archives for SuspensionSim.
2. Show all available CPAR archives, for users who wish to pick and choose specific categories of interest for a new database.

Documentation

The following documents were added:

1. Reference Manuals > VS Embedded Python Interface
2. Technical Memos > Options for Loading Table Data into VS Math Models

The following Reference Manuals were updated:

1. SuspensionSim Commands and Parameters
2. System Parameters in VS Math Models
3. VS Browser Reference Manual
4. VS Commands Reference Manual
5. VS Math Model Manual
6. VS Visualizer Reference Manual
7. VS Output API Manual

The following documents in the **Help** menu were updated:

1. SuspensionSim Model Screens

The following SDK document was updated:

1. The VehicleSim API

Database Examples

Solid Front Axle

New examples have been added to the category/CPAR *** Solid Front Axle**, showing a new steered solid axle suspension template. The example model is shown in ride, roll, and steer tests. This model uses four control arms, a recirculating ball steering linkage, coil springs, and a stabilizer bar.

Screw Axis

New examples have been added to the category/CPAR *** Screw Axis Outputs**, showing how the new screw axis output variables can be used to view the roll, pitch, or steer axis of a knuckle body with VS Visualizer.

SuspensionSim 2023.1 New Features

VS Math Model

Suspension Design Factors (SDFs)

Five new output variables have been added to the list of wheel-based SDFs: wheel travel angle, roll center height, support angle, ride camber rate, and ride toe rate. These are calculated using partial derivatives of the wheel position and orientation with respect to jounce, and they establish instantaneous information about the knuckle's motion at a particular operating point. To support these calculations, there is a new math model keyword for the tire radius used, `RADIUS_SDF`. For more information, refer to the updated **SuspensionSim Commands and Parameters** reference manual.

VS Commands

- SET_OUTPUT_SHORT_NAME was added to allow the manual creation of short (ERD compatible) output names by the user.
- Error checking was added for lengths of new variables and parameters. The name lengths were extended to 48 characters; longer names now generate error messages.
- A new function SAFE_DIV is available for user-defined expressions to avoid divide-by-zero issues.

Other Improvements

- The embedded Python included has been updated to 3.10.2.
- All error messages were reviewed; many were updated to provide more specific information about the cause of the error.

VS Browser

64-Bit Version of the Browser

The browser SuspensionSim.exe is a 32-bit application that runs on both 64 and 32-bit versions of Windows. As such, it can load 32-bit plug-in libraries such as the VS Solver vs_suspsim_32.dll but is not able to use 64-bit libraries.

Most users have been working with 64-bit versions of Windows, and many engineering software tools are now available only as 64-bit applications and libraries. For example, the last version of 32-bit MATLAB from MathWorks was 2015b. That means any recent versions of MATLAB will work only with the 64-bit VS Solver plug-in libraries.

The 2023.1 release includes two versions of the Browser: SuspensionSim.exe (still 32-bit) and SuspensionSim_64.exe (64-bit). The plan from Mechanical Simulation is to drop the 32-bit versions of our tools in a future release. (Recent releases have already included both 32-bit and 64-bit versions of the VS Solver libraries, VS Visualizer, and other tools.)

Mechanical Simulation recommends using the 64-bit version unless there is a need to maintain compatibility with 32-bit tools.

SuspensionSim System Library Screen

The **SuspensionSim System** library screen now sets math model keyword RADIUS_SDF to symbol stack variable <<radius>>. This allows the yellow fields for radius on the **Locations and Bodies** screen or the **SuspensionSim System** screen to be used for the new SDF calculations.

Other Improvements

- Two more plot links were added to the **Generic VS Commands** screen.
- The **Tools** menu was modified to clarify the searching of existing runs for uses of the dataset currently in view.

VS Visualizer

VS Visualizer has added a preferences option to force X or Y plot axis labels to show. Users with a small VS Visualizer window and many plots may have hidden axis labels due to automatic plot window scaling. Forcing the axis labels to show will allow users to view VS Visualizer at their preferred window size.

Licensing

The Command-Line License Manager can now run as a Windows Service, allowing for the application to be started automatically when the system is booted. Additionally, running the License Manager as a Service allows for the application to be started, paused, or stopped using the Microsoft Management Console.

Documentation

The following document was added to the **Help** menu:

- Tools > Database Builder.

The following documents in the **Help** menu have been updated:

- Home: The Run Screen
- Animator > Camera Setup
- Animator > Reference Frames
- Animator > Shapes and Groups
- Generic Data Screens > Data Groups
- Generic Data Screens > Generic Table
- Generic Data Screens > External Parsfile
- Model Extensions > Model Screens
- Model Extensions > Output Variables
- Plot Setup
- Procedures and Events
- SuspensionSim Models
- Tools > Calculator Tool for Tables
- Tools > Preferences
- Tools > VS/ERD File Utility
- Reference Manuals > SuspensionSim Commands and Parameters
- Reference Manuals > System Parameters in VS Math Models
- Reference Manuals > VS API
- Reference Manuals > VS Browser (GUI and Database)
- Reference Manuals > VS COM Interface
- Reference Manuals > VS Commands
- Reference Manuals > VS Commands Summary
- Reference Manuals > VS Math Models
- Reference Manuals > VS Visualizer
- Technical Memos > HPC Licensing
- Technical Memos > The VehicleSim License Manager Utility
- Technical Memos > VS Solver CLI Wrapper
- Technical Memos > Guidelines for Building SuspensionSim Models.

- Release Notes > SuspensionSim Backward Compatibility
- Release Notes > SuspensionSim Database Options
- Release Notes > System Requirements

Database

Updates to Previous Examples

The following examples have been updated to use the new SDF output variables:

- * * Quick Start Guide > Baseline (Ride Test)
- Miscellaneous > COM with Python
- Miscellaneous > Embedded Python
- Multi-Link Example > Ride Test
- RWD Performance > Front Susp. (2) Jounce
- RWD Performance > Rear Susp. (2) Jounce
- SLA Example > SLA w/ Asymmetric Parametric Templates
- SLA Example > SLA w/ Ball-Joint Bushings
- SLA Example > SLA w/ Nonlinear Ride Spring
- SLA Example > SLA w/ Rigid Axial Bushings
- SLA Example > SLA w/ Symmetric Parametric Templates
- SLA Example > SLA without templates
- Strut Example > Strut (Ride Test).

Several minor changes were made to these examples during this process. For example, the **SLA w/ Nonlinear Ride Spring** example has an updated nonlinear spring force table.

New Examples

There is a new example model called **Walking Beam**, representative of what could be used on the rear of a vocational truck with tandem drive axles. The front and rear axles are interconnected by beams which are located by trailing arms. This example model is exercised in three new runs:

- * Walking Beam > A. Design State
- * Walking Beam > B. Jounce Test
- * Walking Beam > C. Roll Test.

SuspensionSim 2022.0 New Features

Database Delivery

In past versions of SuspensionSim, the installer application provided an example database with dozens of examples covering basic and advanced modeling in SuspensionSim, as well as a few examples showing extensions with COM, Python, and MATLAB. For example, the 2021.0 release contained 56 example runs.

That delivery method has been changed for 2022.0. As with all versions over the past 8 years — since SuspensionSim became part of the VehicleSim products — the installer program installs a folder, `SuspensionSim_Prog`, with our complete program stack (VS Browser, SS Solver, VS

Visualizer) and static resources such as animator 3D assets and Help PDF files. However, the installer no longer installs a default database. Instead, 60 examples are provided in 21 Consolidated Parsfile (CPAR) archive files. Once SuspensionSim is installed, users may build databases at any time, using any combination of the CPAR archives.

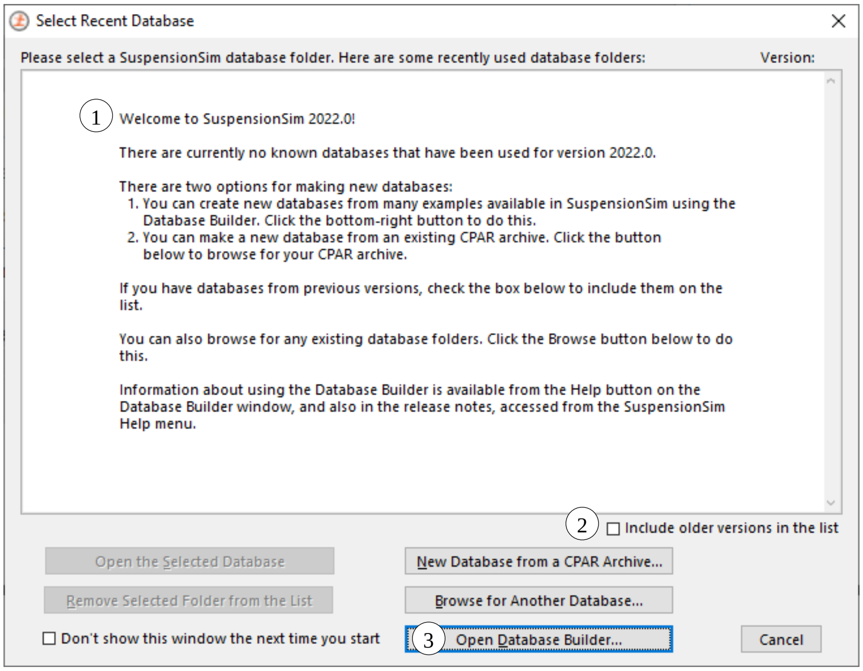
The new tools for installing databases are described in the following section. More information about the specific new datasets for this release is provided in the last section (New Examples, p. 15).

VS Browser

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

Recent Database Window

When opening recent versions of SuspensionSim, a window appears listing database folders that are known for the current version. This window, Figure 1, has been extended for 2022.0, providing access to a new tool: the Database Builder.


Select Recent Database
Please select a SuspensionSim database folder. Here are some recently used database folders:
Version:
1
2
□ Include older versions in the list
Open the Selected Database
New Database from a CPAR Archive...
Remove Selected Folder from the List
Browse for Another Database...
□ Don't show this window the next time you start
3
Open Database Builder...
Cancel

Welcome to SuspensionSim 2022.0!

There are currently no known databases that have been used for version 2022.0.

There are two options for making new databases:

1. You can create new databases from many examples available in SuspensionSim using the Database Builder. Click the bottom-right button to do this.
2. You can make a new database from an existing CPAR archive. Click the button below to browse for your CPAR archive.

If you have databases from previous versions, check the box below to include them on the list.

You can also browse for any existing database folders. Click the Browse button below to do this.

Information about using the Database Builder is available from the Help button on the Database Builder window, and also in the release notes, accessed from the SuspensionSim Help menu.

Figure 1. Select Recent Database window in new version of SuspensionSim.

When launching SuspensionSim 2022.0 for the first time, the **Select Recent Database** window shows a message describing options for building a new database or accessing older databases for use in the new version ①. If you have SuspensionSim database from previous versions, you can check the box **Include older versions in the list** ② and the window will show databases that were used in the previous versions of SuspensionSim. As in past versions, any of those databases may be selected for use in the new version.

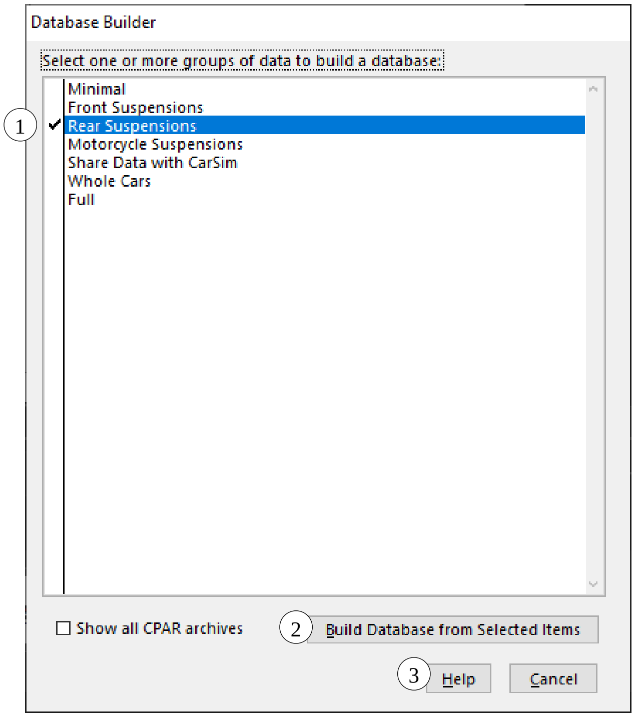
Database Builder

To access examples for 2022.0 when you start SuspensionSim, click the button **Open Database Builder** ③, Figure 1, to bring up the Database Builder, Figure 2. When a group of data is selected ①, the button **Build Database from Selected Items** ② becomes active. The **Help** button ③ opens a PDF document that describes the options. (You can also find this document in the **Help** menu under the **Release Notes** submenu, if you have a database open already.)

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

Figure 2 shows that SuspensionSim 2022.0 provides seven predefined database options. Table 1 provides a little more information about these options.

More information about specific new datasets is provided latter in these release notes, p. 15. For descriptions of all CPAR archives, please see the Help document for the Database Builder.



Database Builder

Select one or more groups of data to build a database:

- Minimal
- Front Suspensions
- ✓ Rear Suspensions
- Motorcycle Suspensions
- Share Data with CarSim
- Whole Cars
- Full

□ Show all CPAR archives

Figure 2. Database Builder window showing only predefined database options.

Table 1. Pre-defined database options.

Database Option	**Description**	**Target Users**
Minimal	Quick Start Guide examples, automation examples (COM/Python/MATLAB), animator shapes, plots, preferences	All users
Front Suspensions	Minimal + standalone car/truck front suspensions	Users focused on front suspension K&C

Table 1. Pre-defined database options.

Database Option	Description	Target Users
Rear Suspensions	Minimal + standalone car/truck rear suspensions	Users focused on rear suspension K&C
Motorcycle Suspensions	Minimal + motorcycle suspensions (front and rear)	Users focused on motorcycle K&C
Share Data with CarSim	Minimal + examples which write data for CarSim	Users focused on getting data into CarSim
Whole Cars	Minimal + front/rear car suspensions which are matched to each other	Users focused on overall vehicle K&C
Full	All the above.	Users who wish to have one database with all examples, like previous versions of SuspensionSim.

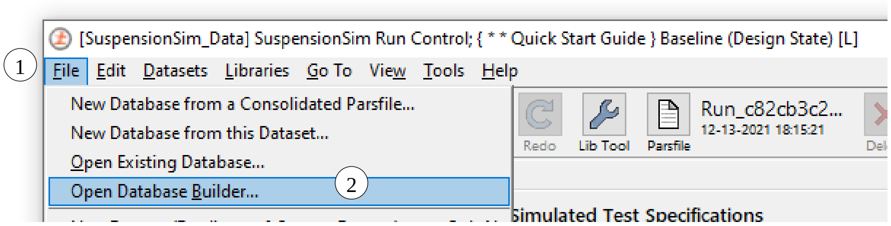
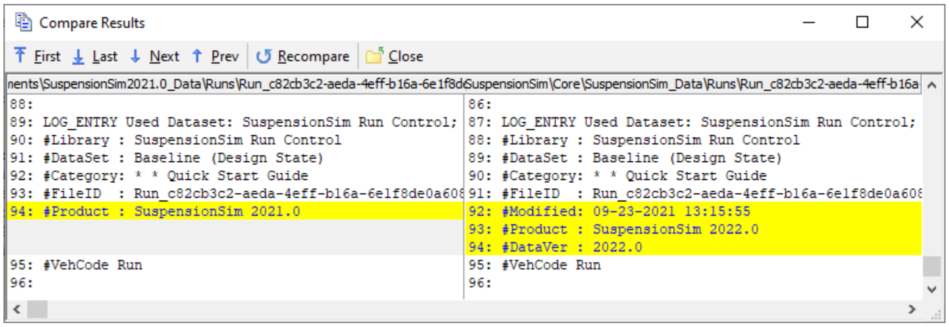

[SuspensionSim_Data] SuspensionSim Run Control; { * * Quick Start Guide } Baseline (Design State) [L]
1
File
Edit
Datasets
Libraries
Go To
View
Tools
Help
New Database from a Consolidated Parsfile...
New Database from this Dataset...
Open Existing Database...
Open Database Builder...
2
Redo
Lib Tool
Parsfile
Run_c82cb3c2...
12-13-2021 18:15:21
Del
Simulated Test Specifications

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

Time and Version Stamps Added to Parsfiles

Parsfiles written by VS Browser are named using a 36-character unique universal ID (UUID), with a prefix associated with the library, for example, `Run_c82cb3c2-aeda-4eff-b16a-6e1f8de0a608` is a name that might be used for a Run Control dataset. This is done to support version control software that tracks creation and revisions of files based on their names.

The Parsfiles always include stamps at the end of the file beginning with '#', such as `#Library`, `#DataSet`, etc. Three new stamps are added in 2022.0 to provide more information about time and version. Two will appear in updated Parsfiles, Figure 4, `#Modified` and `#DataVer`.



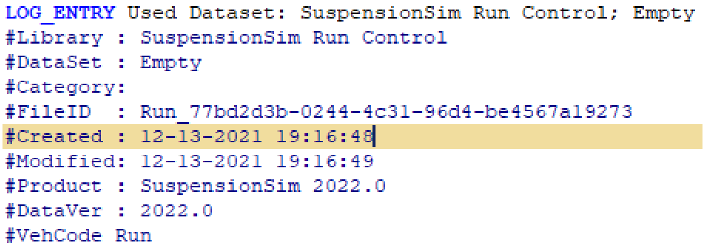
```
nents\SuspensionSim2021.0_Data\Runs\Run_c82cb3c2-aeda-4eff-b16a-6e1f8d  SuspensionSim\Core\SuspensionSim_Data\Runs\Run_c82cb3c2-aeda-4eff-b16a
88:                                                  86:
89: LOG_ENTRY Used Dataset: SuspensionSim Run Control;  87: LOG_ENTRY Used Dataset: SuspensionSim Run Control;
90: #Library : SuspensionSim Run Control             88: #Library : SuspensionSim Run Control
91: #DataSet : Baseline (Design State)               89: #DataSet : Baseline (Design State)
92: #Category: * * Quick Start Guide                 90: #Category: * * Quick Start Guide
93: #FileID  : Run_c82cb3c2-aeda-4eff-b16a-6e1f8de0a608  91: #FileID  : Run_c82cb3c2-aeda-4eff-b16a-6e1f8de0a608
94: #Product : SuspensionSim 2021.0                  92: #Modified: 09-23-2021 13:15:55
                                                     93: #Product : SuspensionSim 2022.0
                                                     94: #DataVer : 2022.0
95: #VehCode Run                                     95: #VehCode Run
96:                                                  96:
```

Figure 4. Parsfile stamps compared for the same Run Control dataset in 2021.0 vs. 2022.0.

`#Modified` identifies the date and time (obtained from Windows) when the file content was last changed. (If the Undo command is used to return a Parsfile to its state at the start of the VS Browser session, the file contents are then fully restored, and `#Modified` will return to the initial value.)

`#DataVer` identifies the version of the database. With all recent releases, this has matched the product version `#Product`, but this has not always been the case, and could be useful in the future.

The third new stamp is `#Created` which is another timestamp obtained from Windows. This will appear in Parsfiles created using SuspensionSim 2022.0, rather than files which were updated from previous versions. This stamp will never change even after updates to versions beyond 2022.0.



```
LOG_ENTRY Used Dataset: SuspensionSim Run Control; Empty
#Library : SuspensionSim Run Control
#DataSet : Empty
#Category:
#FileID  : Run_77bd2d3b-0244-4c31-96d4-be4567a19273
#Created : 12-13-2021 19:16:48
#Modified: 12-13-2021 19:16:49
#Product : SuspensionSim 2022.0
#DataVer : 2022.0
#VehCode Run
```

Figure 5. A Parsfile snippet, created in 2022.0, showing the #Created stamp.

Find Links to Current Dataset from Selected Library

A new command was added to the **Tools** menu for searching: **Find All References to this Dataset in "Run_all.par" Files, by Library**.

The existing **Find All References to This Dataset** option finds datasets that are immediately referenced by others. The search can be repeated, to find datasets that reference datasets that reference the current dataset, but this form of manual repeating is time-consuming.

The new search option takes advantage of `Run_all.par` files that are made automatically whenever a run is made, or VS Visualizer is used to view results. This has significant advantages:

2. Searching these files is rapid compared to searching the entire database. Even with large databases, the search is much less than a minute.

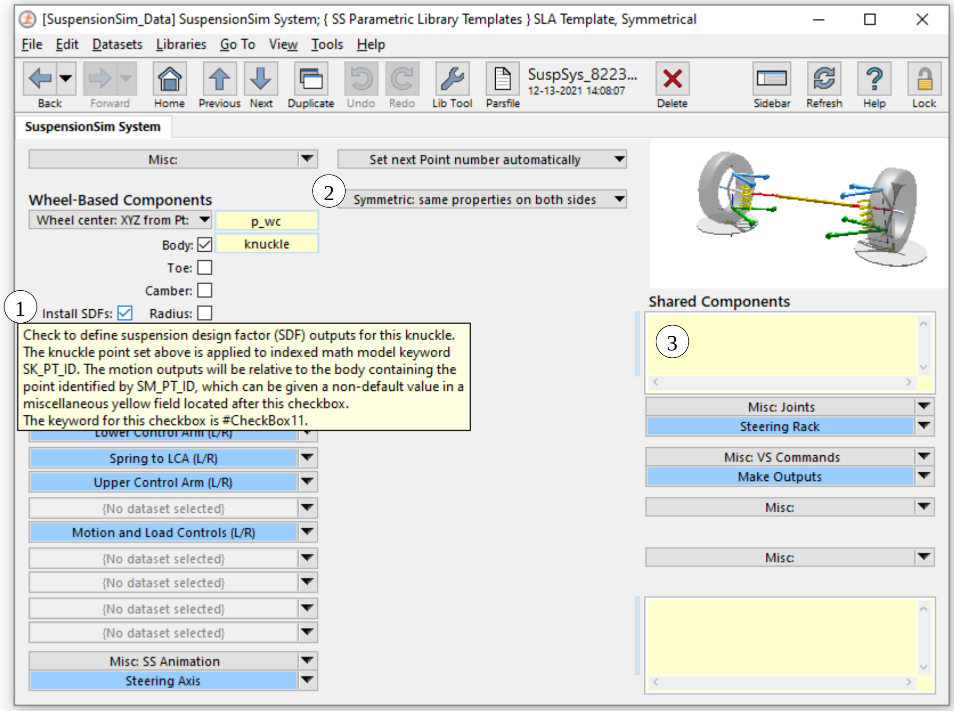
3. The searching finds all references where the current dataset was used in an existing run.

4. Results are filtered to show only cases where the current dataset is referenced (no matter how indirectly) by datasets in a specified library.

For example, perform a search from a SuspensionSim System dataset to find all SuspensionSim Locations and Bodies datasets which use the same system topology. Or, find all SuspensionSim Run Control datasets which use a particular dataset.

GUI Support for SDF Outputs

The SuspensionSim System library screen has a new checkbox in the **Wheel-Based Components** column, **Install SDFs** ①, Figure 6. The checkbox installs outputs for the left, right, or both wheels, depending on the symmetry options ②. This checkbox can be used to activate the new SDF output variables described in the following section.



Wheel center: XYZ from Pt: p_wc

Body: ☑ knuckle

Toe: □

Camber: □

Install SDFs: ☑ Radius: □

Check to define suspension design factor (SDF) outputs for this knuckle. The knuckle point set above is applied to indexed math model keyword SK_PT_ID. The motion outputs will be relative to the body containing the point identified by SM_PT_ID, which can be given a non-default value in a miscellaneous yellow field located after this checkbox.
The keyword for this checkbox is #CheckBox11.

Figure 6. SuspensionSim System screen showing tool tip for the new Install SDFs checkbox.

SS Solver

SS Solver is the SuspensionSim solver program which reads input files, performs all calculations in the multibody suspension model, and generates output files that are viewed or shared with other software. SS Solver is a VS Solver which was extended to perform equilibrium calculations for a model built using SS Commands and parameters created by those commands. Version 2022.0 introduces a new output type to SS Solver, called Suspension Design Factors (SDFs).

Suspension Design Factors

SDFs in general are measurements commonly needed during the design or evaluation of a suspension. For example, the camber angle of a wheel relative to its vehicle as a function of wheel center vertical displacement. Previous versions of SuspensionSim (since it became a VehicleSim product) relied extensively on user definitions for these kinds of output variables, using the VS Command `DEFINE_OUTPUT` and raw multibody output variables produced by SS Solver, such as global point coordinates or Euler angles. Now many SDFs exist natively in SS Solver and are available to activate with the SS Command `SK_PT_ID`:

`SK_PT_ID(`*iaxle*`,`*iside*`)` *ipoint* defines SDFs for the wheel on axle *iaxle* located by *iside*. The *ipoint* argument must give the point on the steering knuckle body corresponding to the wheel center.

For example, command `SK_PT_ID(1,1)` defines the front left wheel outputs seen in Table 2.

Table 2. SDF outputs for front left wheel (iaxle=1, iside=1).

Wheel SDF	Units	Description
`Cam_L1`	deg	Angle between the wheel plane and the sprung mass Z axis, negative if the top of the wheel leans toward the sprung mass.
`Div_L1`	deg	Change in orientation of the steering knuckle about the wheel spin axis. This angle, together with toe and camber, defines the total orientation of the steering knuckle relative to the sprung mass. Positive dive is a counter-clockwise rotation when viewed from the left side.
`Inc_L1`	deg	Left wheel inclination = -camber. Right wheel inclination = camber.
`Str_L1`	deg	Left wheel steer = -toe. Right wheel steer = toe.
`Toe_L1`	deg	Angle between the sprung mass X axis and the line of intersection of the wheel plane and the sprung mass X-Y plane. Positive if the wheel is steered inward (toe-in).
`WCx_L1`	mm	X displacement, positive forward, of point on steering knuckle corresponding to wheel center, using the sprung mass coordinate system.
`WCy_L1`	mm	Y displacement, positive inward, of point on steering knuckle corresponding to wheel center, using the sprung mass coordinate system.
`WCz_L1`	mm	Z displacement, positive upward, of point on steering knuckle corresponding to wheel center, using the sprung mass coordinate system.

The `SK_PT_ID` command can be used up to four times, suitable for a four-wheel vehicle model. The first time the command is used, a parameter `SM_PT_ID` is created. This parameter may be used to identify a sprung mass point, useful if the model has a sprung mass body which is not equivalent to the ground body. (In most models, the default value for `SM_PT_ID` suffices.)

If the `SK_PT_ID` command is used twice on one axle, for each side, SS Solver defines axle outputs. For example, Table 3 shows the axle-specific outputs available when the commands `SK_PT_ID(1,1)` and `SK_PT_ID(1,2)` have both been used.

Table 3. SDF outputs for iaxle=1 (both sides have been activated).

Axle SDF	Units	Description
`MPx_A1`	mm	X displacement, positive forward, of the midpoint of the line through the wheel centers, using the sprung mass coordinate system.
`MPy_A1`	mm	Y displacement, positive leftward, of the midpoint of the line through the wheel centers, using the sprung mass coordinate system.
`MPz_A1`	mm	Z displacement, positive upward, of the midpoint of the line through the wheel centers, using the sprung mass coordinate system.
`Rol_A1`	deg	Roll angle of the line through the wheel centers, using the sprung mass coordinate system. Positive when the right wheel moves up toward the sprung mass and the left wheel moves down.
`Trk_A1`	mm	Length of the line between the left and right wheel centers.
`Yaw_A1`	deg	Yaw angle of the line through the wheel centers, using the sprung mass coordinate system. Positive when the axle steers to the left.

In VS Browser, the checkbox described in a previous section, p. 10, may be used easily to automatically write the correct `SK_PT_ID` SS Commands, as this library screen already has access to the correct *ipoint* argument as well as the *iaxle* and *iside* context. If `SM_PT_ID` needs a non-default value, it can be set anywhere after the checkbox, such as the miscellaneous yellow field on the SuspensionSim System screen ③, Figure 6.

For any model with SDFs defined, all SDF outputs can be viewed using the **View** controls on the bottom right of the Run Control screen. Select **Outputs from math model (text)** and search for SDFs.

For more on how SS Solver defines and calculates SDFs, see the *SuspensionSim Commands and Parameters Reference Manual*. For more on building models which use SDFs, see the *Guidelines for Building SuspensionSim Models Technical Memo*.

> **Note** SDFs assume the ISO/SAE axis convention (X forward, Y left, Z up) and may not produce meaningful output if another axis convention is used when building the model.

Miscellaneous

A new system parameter `OPT_ECHO_DEFAULT` is available to disable the identification with the indicator [D] in the Echo file for default values that were not set by reading an input file. This can be helpful in advanced applications, such as when a new run is made using an Echo file written at the end of a previous run. In this case, the [D] indicator never appears in the Echo files for the continuation run. The new parameter may be used to ensure the [D] indicator doesn’t appear in other runs either, simplifying the use of text editors to compare files.

VS Visualizer

VS Visualizer can now compute first and second derivative estimates of data channels (with respect to time) and use these for animation and plotting. See the VS Visualizer reference manual for more.

VS Visualizer has a new option to scale plots non-symmetrically by pressing the Alt key while dragging with the middle-mouse-button (or left+right mouse buttons). In this mode, moving the mouse up/down zooms vertically, moving left/right zooms horizontally. There is also an option on the plot window right-click context menu, “Asymmetric mouse zoom”, to switch which mode is default (active w/o the Alt key).

Documentation

The **Table of Contents for Screens** has been updated.

The following **Screen Documents** have been updated:

5. Animator: Reference Frame
6. Animator Shapes and Groups
7. Batch Matrix
8. Calculator: Symbolic
9. Generic Data Screens
10. Generic Table Screen

11. Setting up Import and Output Variables
12. External Models and RT Systems
13. VS Browser Preferences
14. Run Control Screen (Home)
15. SuspensionSim Model Screens
16. Writing Custom Parsfiles from SuspensionSim

The **Quick Start Guide** has been updated.

The following **Reference Manuals** have been updated:

17. SuspensionSim Commands and Parameters Reference Manual
18. System Parameters in VS Math Models
19. The VehicleSim API: Running and Extending VS Math Models
20. VS Browser Reference Manual
21. VS COM Interface: API of the VS Browser
22. VS Commands Reference Manual
23. VS Commands Summary
24. VS Math Models Reference Manual
25. VS Visualizer Reference Manual

The following **Technical Memos** have been updated:

26. Licensing for High-Performance Computing (HPC)
27. Borrowing a VS Product License from a FlexLm Network Server
28. Guidelines for Building SuspensionSim Models
29. Automating Runs with the VS API
30. Command Line Vehicle Sim License Manager
31. VS Solver Wrapper

The following **Release Notes** have been updated:

32. Terms and Conditions of Sale and EULA
33. Network Licenses for VehicleSim Products
34. Silent Installation for VehicleSim Products
35. SuspensionSim Backward Compatibility
36. SuspensionSim 2022.0: New Features
37. VehicleSim System Requirements & Software Compatibility

38. Technical Support

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

39. Release Notes > SuspensionSim Database Options

Database

Updated Examples

Nearly all existing examples have been updated to use the new SDF outputs for camber, toe, and dive angles and wheel center X, Y, and Z displacements (see p. 11). Accordingly, a large number of VS Commands outputs, created with `DEFINE_OUTPUT`, have been removed from the examples. Plots have been updated to refer to the new native output variables, rather than the previous custom output variables. This allowed various **Plot: Setup** datasets to be deleted, as output names became common where before they were often example specific.

As part of this process, the results for all examples were plotted side by side to confirm that the new native output variables reproduced the previous VS Commands custom definitions.

Due to the scope of these changes, be careful, or try to avoid, importing old versions of the SuspensionSim examples into a database containing the new versions. There is no backwards compatibility issue when creating a new database from an older CPAR or updating an old database to the new version; the conflict would arise if any old data with the same UUIDs is mingled with the new data — and the CPAR import process would warn of any conflicts. (The outputs themselves are never active unless the **Install SDFs** checkbox or `SK_PT_ID` SS Commands are manually specified.)

The "Whole Car" example model has been updated to use a new animation shape, Figure 8.

New Examples

There is a new example of a complete car available in the Whole Cars or Full database option (see p. 8 for more on database options). The individual CPAR is called

```
-_Open_Wheel__Sports_Car.cpar
```

and creates a category called *** Open Wheel / Sports Car**. This category contains a series of examples exercising a model of an open wheeled racing car, Figure 7. The **Continue from a previous run** and **Events** features are used extensively. This is an advanced model and familiarity with SuspensionSim is assumed.

New Animator Resources

Two new 3D assets have been provided in the Animator: Shape File Link library:

- SS Library > Car
- SS Library > Truck.

These are shown in Figure 8 and Figure 9, respectively.

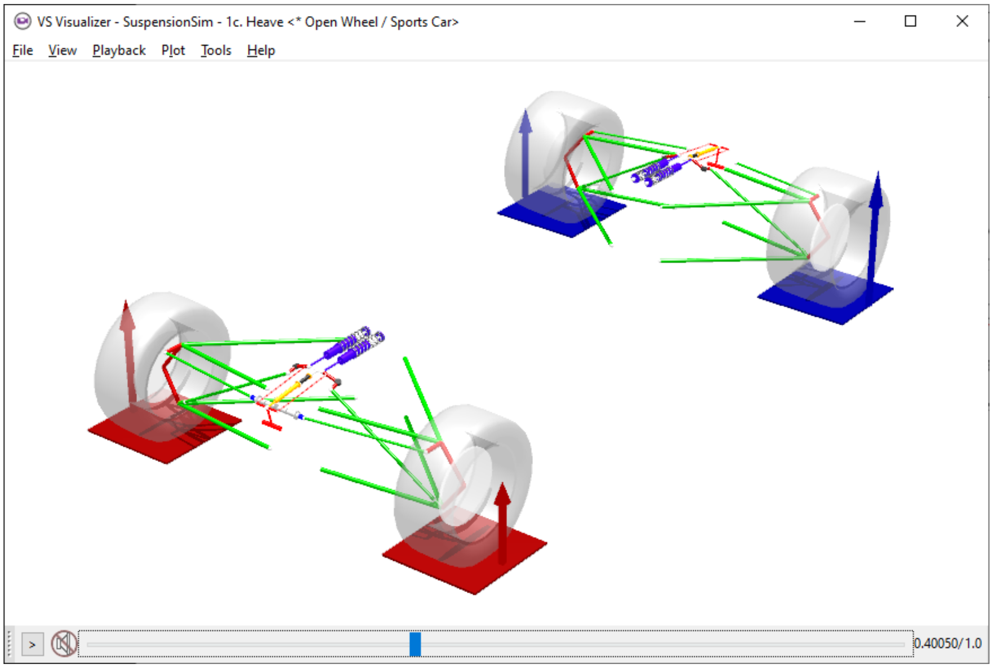

VS Visualizer - SuspensionSim - 1c. Heave <* Open Wheel / Sports Car>
File
View
Playback
Plot
Tools
Help
>
0.40050/1.0

Figure 7. Animation output for the new Open Wheel complete car.

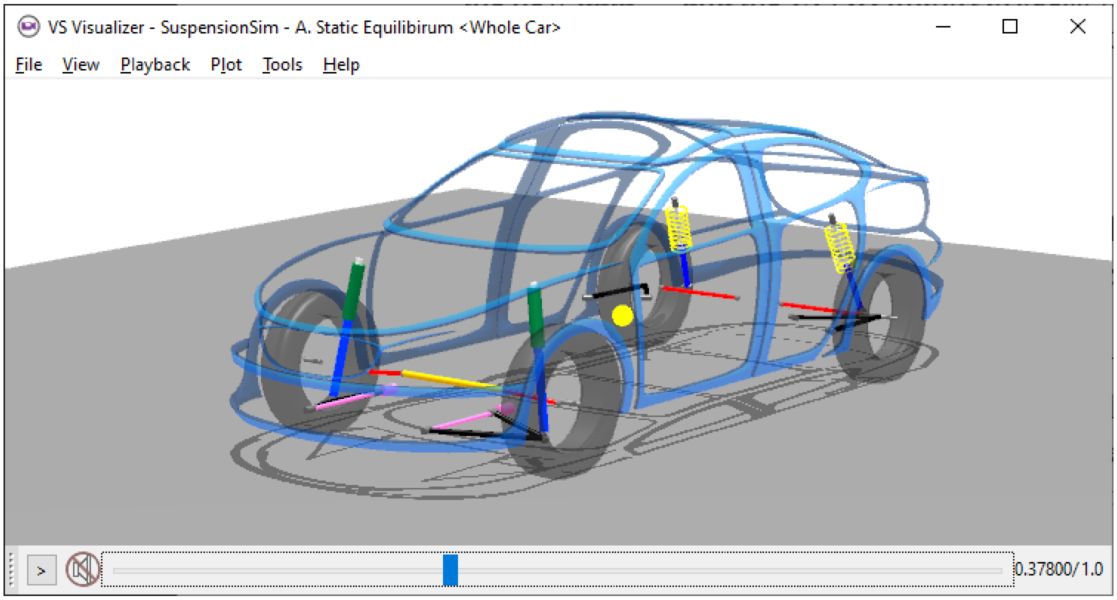

VS Visualizer - SuspensionSim - A. Static Equilibirum <Whole Car>
File
View
Playback
Plot
Tools
Help
>
0.37800/1.0

Figure 8. Updated "Whole Car" example, using the new car shape.

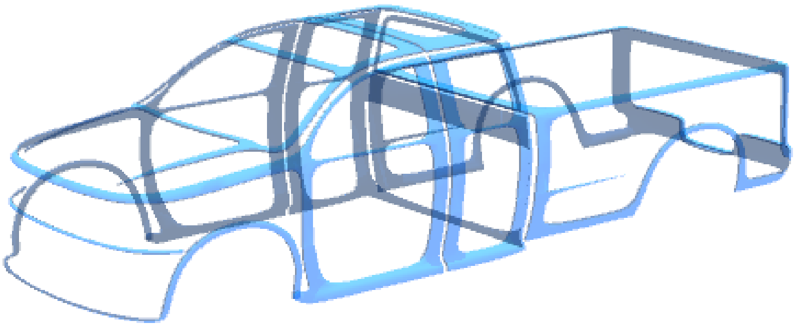


Figure 9. The new truck shape.

SuspensionSim 2021.0 New Features

The SuspensionSim Installer

The SuspensionSim installer has been reconfigured to package some of the installation files separately from the executable. The executable has the same appearance as previous versions:

`Setup_SuspensionSim_2021.0_r`*`rev`*`.exe`

where *rev* is a revision number. As before, the installer may later be used multiple times to create new, clean `SuspensionSim_Data` folders. With the new installer format, rather than saving just the executable for this purpose, you should retain the ZIP or ISO file which contains the executable and the additional installation files.

Architecture Changes

A revision to the scaling of animator shapes was made so that the supporting code is now shared across all Mechanical Simulation products. This change requires action to use some old datasets or databases. See the SuspensionSim Backward Compatibility document for more information.

Model Features

Minor improvements have been made to the parsing of VS Commands. These should be transparent to most users.

VS Browser: Graphical User Interface (GUI)

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

- A new option **Find All Deprecated Assets** has been added to the VS Browser under the **Tools** menu. This will produce a list of every dataset in the current database that has a reference to a deprecated animator asset. (Mechanical Simulation routinely deprecates and then retires 3D shapes used in VehicleSim products such as CarSim. As of SuspensionSim 2021.0, no SuspensionSim animator assets have been deprecated or retired by Mechanical Simulation, so none are expected to be found.)

- A new option **Re-Write every 'All Parsfile' sent to the Solver or VS Visualizer** has been added to the **Tools** menu. This creates the Run_all.pars which are found in the Results folders, but does not make the runs, useful for updating run data when controlling the run itself from outside of the VS Browser.

VS Visualizer

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

New and Updated Examples

Several new examples are included in the 2021 database:

1) The category "* SuspensionSim 2021.0 - Whole Car" shows how an entire car, consisting of a sprung mass and front and rear independent suspensions, can be modeled. Four test cases are included:

    a. Static equilibrium. The sprung mass is allowed to settle into an at-rest condition.

    b. Heave. The sprung mass is moved vertically, exercising the jounce DOFs of the suspensions.

    c. Roll. The sprung mass has a roll moment applied about its X-axis, producing a roll of the sprung mass relative to the ground plane.

    d. Steer. A static steer test produced by translating the steering rack body.

2) The category "* SuspensionSim 2021.0 - 4 Link w/ Track Bar" contains the example "Compliance Test". This can be used to generate compliance coefficients for a CarSim or TruckSim solid axle suspension.

3) The category "* SuspensionSim 2021.0 - RWD Performance" contains eight examples to characterize both suspensions for a sports car. These examples were used to generate the K&C data for the CarSim 2020.1 (and up) virtual steering axis B-class sports car.

    a. Front Susp. (1) Design Configuration

    b. Front Susp. (2) Jounce

    c. Front Susp. (3) Steer

    d. Front Susp. (4) Jounce & Steer

    e. Front Susp. (5) Compliance

    f. Rear Susp. (1) Design Configuration

    g. Rear Susp. (2) Jounce

    h. Rear Susp. (3) Compliance

Documentation

The following help files have been updated:

- Run Control Screen (Home)
- Generic Data Screens
- Model Screens
- Output Variables
- Procedures and Events
- Preferences
- Quick Start Guide

The following reference manuals have been updated:

- System Parameters in VS Math Models, renamed from System Parameters in VS Solvers, including SuspensionSim-specific improvements to better identify system parameters and commands which are applicable to other VehicleSim products but not available in SuspensionSim.
- VS Commands Reference Manual, including a few SuspensionSim-specific notes about VS Commands not applicable to SuspensionSim.
- VS API
- VS Browser
- VS Visualizer
- VS Math Models (former title: VS Solver Programs)

The following document has been added to the **Help** menu in the **Reference Manuals** category:

- VS Commands Summary, a two-page quick reference sheet for VS Commands

The following technical memos have been updated:

- Automating Runs with VS API

The following release notes have been updated (excluding this document):

- SuspensionSim Backward Compatibility
- System Requirements

SuspensionSim 2020.0 New Features

The SuspensionSim Installer

After the initial installation of SuspensionSim, the installer

```
Setup_SuspensionSim_2020.0_rrev.exe
```

(where *rev* is a revision number) may later be used multiple times to create new, clean `SuspensionSim_Data` folders. When used to install a new `SuspensionSim_Data` folder, if a folder is selected that is not empty, the installer will make a new one with the same root name, using the Windows convention of adding a suffix to indicate the name was already used. For example, if an existing folder named `SuspensionSim_Data` is selected, the installer will create a new one named `SuspensionSim_Data (1)`.

A warning about needing administrator permission was changed to clarify that such permissions are not needed to install a new database folder.

Architecture Changes

COM Interface

Two new COM functions have been added: `Run_Background` and `StopWindowsRun`:

- `Run_Background` allows an end-user to run a simulation in the background,
- `StopWindowsRun` allows an end-user to stop the simulation.

Embedded Python

Python 3.6.5 (32-bit and 64-bit) has been embedded in the VS Solvers. New commands allow users to take advantage of Python to further access VS Table information, as well as some improved debugging capabilities.

VS Commands

New Boolean Operators for VS Commands

New infix operators for Boolean operations (&, |, <, >, <=, >=, ==, ~=) have been added to VS commands so that conditional expressions can be more easily assembled. The existing routines `GT()`, `AND()`, etc. remain supported.

A new `IF`(*x*, *y*, *z*) "special function" was introduced for use in formulas. It differs from other functions by only evaluating two of the three arguments. The argument *x* is evaluated and if *x* not equal to *0*, the argument *y* is evaluated and returned as the value of the function; on the other hand, if *x* is equal to *0*, the argument *z* is evaluated and returned as the value of the function. This new function, along with the new User-defined function capability, allows for the conditional processing of different groups of expressions during a simulation.

Improvements for VS Events

The syntax for the `define_event` command has been simplified to have just two arguments, with the second being optional:

`DEFINE_EVENT` *formula [pathname]*

The reporting of Events in the Log file has been modified to more closely match the appearance of the Events as shown in the Event GUI screen. The *formula* argument can be a complicated Boolean statement, or just a variable name. It may even be a constant, e.g., 1, to indicate that the Event should unconditionally be triggered at the next time step.

The previous syntax for `DEFINE_EVENT` has been retained for a new command `MAKE_EVENT`:

`MAKE_EVENT` *variable operator reference [pathname]*

This provides backward compatibility for old datasets in which *reference* is a number that automatically converted from user-units of *variable* to internal units. It also provided support for the old `DELETE_EVENTS` command.

User-Defined Functions

Users can now define new functions with optional arguments, local variables, and return value. The new functions are started with the `begin_function` command, and end with the `end_function` statement. These can process a series of equations.

New and Updated Examples

- A leaf spring rear suspension using the three-link approximation method has been added. For full details of the three-link approximation method, see the new technical memo *Leaf Spring Suspensions in SuspensionSim*, available from the **Help** menu.
- A recirculating ball steering linkage example has been added. This models from the Pitman arm to the roadwheels, specifically a *Haltenberger linkage*.
- An example using embedded Python to calculate output variables is now available.
- Automation of the SuspensionSim GUI using COM from external Python is now demonstrated using a new example.
- A new example generates files for the external control of SuspensionSim's solver. An included MATLAB script can use these to control the solver via the VS API.
- The Batch Runs dataset for All Runs has been updated to include all examples.
- Twenty-eight dataset names have been shortened so that their full text is visible when linked from a SuspensionSim screen.

Documentation

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

- *Table of Contents for Screens*, listing and linking to all help files associated with specific SuspensionSim screens.
- *Joint Rotations and Bushings in SuspensionSim*, a technical memo providing insight into how SuspensionSim represents spatial rotations and suspension bushings.
- *Leaf Spring Suspensions in SuspensionSim*, a technical memo giving details of the three-link approximation used in SuspensionSim's leaf spring rear suspension example.

The following technical memos have been updated:

- Guidelines for Building SuspensionSim Models

- SuspensionSim Commands and Parameters
- VS Solver Wrapper

The following screen documents have been updated:

- Animator: Camera Setup
- Animator: Reference Frame
- Animator: Shapes and Groups
- Plot Setup
- Procedures and Events
- Run Control Screen (Home)
- SuspensionSim Models

The following reference manuals have been updated:

- System Parameters in VS Solvers
- VS API
- VS COM Interface
- VS Solver Programs
- VS Commands

The following release notes have been updated (excluding this document):

- SuspensionSim Backward Compatibility
- System Requirements
- Version 9 Backward Compatibility: Database and Automation

SuspensionSim 2019.0 New Features

The SuspensionSim Installer

After the initial installation of SuspensionSim, the installer `Setup_SuspensionSim_2019_r`*`rev`*`.exe` (where *rev* is a revision number) may later be used multiple times to create new, clean `SuspensionSim_Data` folders. When used to install a new `SuspensionSim _Data` folder, the installer for 2019.0 is a little friendlier to use than in past releases: if a folder is selected that is not empty, the installer will make a new one with the same root name, using the Windows convention of adding a suffix to indicate the name was already used. For example, if an existing folder named `SuspensionSim_Data` is selected, the installer will create a new one named `SuspensionSim_Data (1)`.

A warning about needing administrator permission was changed to clarify that such permissions are not needed to install a new database folder.

Architecture Changes

Longer Names for Output Variables

The VS output file format was introduced in 2014 as an alternative to the legacy ERD file format. In version 2017.0, support was added for comma-separated variable (CSV) text files that can be opened by Excel and other spreadsheet programs. ERD files restrict the short variable names used as keywords to 8 characters; CSV and VS files support longer names.

The VS Solvers in the 2019.0 release make use of longer names to provide consistent naming of variables when there are many instances. If the output variable has a name longer than 8 characters, VS and ERD output files include both the long name and a name limited to 8 characters. VS Visualizer and the VS/ERD File Utility now recognize both the `SHORTNAM` and the `NAME` keywords when reading ERD files. Therefore, datasets for plots and animation from older versions that specify names with the 8-character limit are processed correctly.

The keyword `SHORTNAM` has been used since the ERD format was published in 1984. Because the `SHORTNAM` information is still written to the ERD header, legacy third-party utility programs that support the ERD format should still operate correctly.

Documentation files that list outputs (as obtained with the **View** button on the **Run Control** screen) show both full and short names when the full name has more than 8 characters, as do the VS Browser screens used to view and select outputs, such as **Plot Setup**, **I/O Channels: Export**, and **I/O Channels: Write**.

Error and Range Checking in VS Commands Functions

Formulas used in VS Commands and other inputs can make use of about 45 built-in functions. In past versions, functions such as `sqrt`, `pow`, `sin`, `cos`, were handled by passing the arguments directly to native C library functions (as declared in `math.h`). A few of these functions have range limits. For example, the function `asin` requires that the argument be within ±1; the function `sqrt` requires that the argument not be negative.

In version 2019.0, two levels of range checking are added. Functions that have known ranges now check the argument. If it goes outside the range, the argument is changed to the limit and then the function is applied. For example, `asin(-1.01)` is evaluated as `asin(-1.0)`. As another example, `sqrt(-0.123)` is evaluated as `sqrt(0.0)`.

If results show a math exception (`nan` or `inf`, indicating "not a number" or "infinite," respectively), an error is generated, and the formula is shown.

In past versions, the formula in the VS Command would return a `nan` or `inf`, but there would be no reporting. In many cases the VS Commands were not critical, so users were unaware of the issue.

Miscellaneous

1. New system constants were added and are visible in the Echo file: `DR`, `G`, `PI`, and `ZERO`. The constants `DR`, `G`, and `PI` existed in past versions but were not listed in the Echo file. The constant `ZERO` is provided to simplify the use of VS Events when evaluating formulas as being either zero or not zero.

2. A new system parameter was added: `OPT_ECHO_KEYWORDS_LC`. When set to a nonzero value, all keywords and commands listed in the Echo file are written in lowercase. This is helpful when copying equations or keywords to paste into Miscellaneous yellow fields, given that lowercase words take up less space.
3. Table entries of type `STEP` can now be updated after definition or loading with an equation. (e.g., `eq_init example_tab(0,1,1) = 10`).
4. Attempts to set units of Configurable Functions with the VS Commands `SET_UNITS_TABLE`, `SET_UNITS_TABLE_COL`, and `SET_UNITS_TABLE_ROW` would perform no action if the keywords for the table function or units were not recognized. These now generate errors telling which keyword was not valid.

Model Features

VS Commands

The VS Commands Reference Manual has been updated.

A new VS Command function has been added — `GENSEED` — which generates a non-repeatable seed number for the existing random number function `SRAND`. The `GENSEED` function is based on the `_rdtsc()` and `RDTSC IA-32` processor instruction. This command retrieves the number of clock cycles that have occurred on the CPU since it booted up.

VS Event Group ID

Two new VS Commands were added to help manage pending Events. Newly created Events are assigned an ID number. The default ID is zero, but it can be changed with the new command `SET_EVENT_ID`. Another new command `DELETE_EVENTS_ID` deletes all pending events with the specified ID, while leaving other Events intact. This allows groups of Events to be managed independently of each other, e.g., to control traffic vehicles with sets of Event independent of those used for the ego vehicle.

The Echo file section that lists Events now includes `SET_EVENT_ID` commands as needed to reproduce the IDs for each pending Event.

Miscellaneous

1. A new section was added to the Echo file showing constants (`DR`, `G`, `PI`, and `ZERO`).
2. A few parameters are locked once a simulation starts. Most of these involve file i/o or numerical integration (`TSTEP`, `IPRINT`, etc.). They are now identified in the Echo file with a `[L]` written at the end of the parameter description.
3. Constants and calculated variables that should not be changed cannot be changed using a normal assignment or VS Commands such as `EQ_IN`. Trying to do so will generate an error message.
4. Equations added with VS Commands for initialization (`EQ_INIT`, `EQ_INIT2`, and `EQ_PRE_INIT`) are removed after the initialization, such that they do not appear in the End Echo file that might be used to continue the simulation with a new run.

5. Integer variables that act as switches for more than two options are now checked for a minimum and maximum values (e.g., `OPT_INT_METHOD`, `OPT_TIRE_MODEL`). Integer switches with only two values are not checked; they work as 0 or not 0.

6. Import variables added with the `DEFINE_IMPORT` command are checked to ensure the new names begin with the prefix "`IMP_`". Also, variables added with the command `DEFINE_PARAMETER`, `DEFINE_OUTPUT`, and `DEFINE_VARIABLE` are checked to ensure that they do not begin with the prefix "`IMP_`".

Graphical User Interface (GUI) and Database

Run Control and Procedures Screens

The **Run Control** and **Procedures** screens include lists of **Plot: Setup** blue links that were fixed to the **Plot: Setup** library in past versions. Both have been modified to allow groups of **Plot: Setup** datasets to be specified.

Two of the links on the **Procedures** screen have been extended to allow generic libraries to be used. Datasets in the Generic libraries can in turn be linked to multiple **Plot: Setup** datasets. For example, the **Generic VS Commands** screen has 16 links that can be used for **Plot: Setup** datasets. On the **Procedures** screen, the first and tenth links have been extended. Whatever library is used for the first link will also be used for the next nine links. Similarly, whatever library is used for the 11th link will also be used for the next eight.

The **Run Contro**l screen has less available space, so only one of the links is extended: Link 6. Links 7 and 8 are set to whatever library is used for Link 6. Links 1-5 remain fixed to the **Plot: Setup library**.

Preferences Screen

The **Preferences** screen was modified to fix some existing problems and support new features such as the increased number of animation resources. Past versions of the screen included a few blue links for default datasets to be used for Plot Formats. These links led to problems in some cases when transferring data with CPAR files, so they were removed for version 2019.0. Here are some changes that were made:

1. A checkbox was added to support a new option: "Perform batch run updating when deleting or renaming Run screen." This is helpful for users who make batch runs with the **Batch Runs** tool. When this is active and a dataset from the **Run Control** screen is deleted or renamed, the Browser scans all datasets in the **Batch Runs** library and fixes invalid references to the old name

2. A **Secondary Resources** field was added for specifying a location for VS Visualizer to find additional assets. Users can now add a folder with custom assets.

3. A checkbox for selecting the deprecated tool WinEP was removed; SuspensionSim 2019.0 does not support it.

4. The blue link for a default plot format has been removed. New versions have links to a plot format for every **Plot Settings** dataset. Imported plot datasets without a specified plot format will use the default from VS Visualizer.

5. A new checkbox control was added to the screen to disable the writing of timestamp information in exported Parsfiles. This was done because new timestamps can cause confusion with some version control software.

6. A new drop-down control was provided to choose between 32/64-bit versions of VS Visualizer.

7. Right-click information for controls was updated to better describe the behavior in 2019.0.

Events Screen

The **Events** screen was modified to include two new options for the simple cases of "If a formula is not zero" and "If this formula is zero".

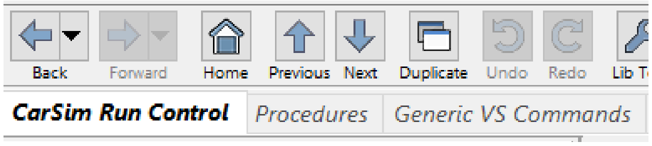
1. A new system parameter, ZERO, was added to the VS Solvers. This is the variable to monitor, with a ~= operator, and a symbolic expression that will trigger the Event when not zero.

2. Two more options were added on the Event drop-down control: If, and If Not. These use the existing ~= and ==.

3. The yellow fields for formulas are slightly larger, and now include resize controls on the right edge so they can be stretched to view longer formulas.

Library Tabs for Rapid Access to Screens

The VS Browser has a set of tab controls located beneath the toolbar that can be used to quickly navigate between libraries that were recently visited.



Note that switching to a different dataset displayed under a pinned tab does NOT affect the dataset linkage. Linkage is supplied by blue links only.

Searching the Database

An option has been added to the search initiated with the **Tools** menu item **Find Text in the Database** that limits the search to the current dataset and all linked datasets.

COM Interface

Two new COM functions have been added: `CreateAnimatorParFile` and `CreateAllParFile`. These functions allow an end-user to create an `animator.par` file and `run_all.par` file, respectively, providing the capability to generate these files without the need to run a new simulation. Running a new simulation will still generate the `animator.par` and `run_all.par` files automatically, as well as new simulation results.

The VS COM interface has been expanded to include functions that can return error information. For example, a batch run may return an error message for a specific run that failed. These new functions end in `_CheckError`. A complete list of functions can be found in the VS COM Interface technical memo.

Miscellaneous

1. It is now possible to cache any number of run results (not just multiples of five) for a given run. On the Run Control screen, select “Keep N” using the drop-down menu and input a custom value using the yellow field which appears to the right of the control.
2. A new command line option has been added to assist in automation. Adding the `/SETQUICKLAUNCH` option when running from the command line will disable the License Manager and Database Selection dialogs from appearing at startup.
3. Many of the miscellaneous yellow fields are now resizable to provide a quick view of the contents without scrolling.
4. The **Plot Setup** screen was modified to remove a checkbox that only applied for the obsolete WinEP plotter, and to add support for the 1:1 X-Y scaling that is now supported by VS Visualizer.

VS Visualizer

1. VS Visualizer is now available as a 64-bit version, supporting large simulations with binaries exceeding 4GB of data. The VS Visualizer platform can be changed between 32-bit and 64-bit from the SuspensionSim **Preferences** screen, complementing the 32-bit version included with previous releases.
2. In support of more automated modes of operation, along with the COM commands `CreateAnimatorParFile` and `CreateAllParFil`e, a new command line parameter has been added which allows end-users to generate VS Results Analysis Package (VSRAP) files. The new parameter `-vsrap` “<filename>” allows an end-user to build a VSRAP file from the control file `animator.par`; when the command line option is used, VS Visualizer will load the specified Parsfile, convert it to a VSRAP file, and then close automatically. Please see the VS Visualizer documentation for more details regarding this new option.
3. A new option **Export Video** has been added (available from the **File** menu) to create an AVI (audio-video interleaved) file that can be dragged into Microsoft PowerPoint® or viewed with Windows Media Player and other video players.
4. An option has been added to specify the same scaling for the X and Y axes. This is enabled with a checkbox on the **Plot Setup** screen. This is useful for plots of trajectories of points on the suspension.
5. When VS Visualizer is launched by clicking the **Plot** button on the **Run Control** screen, it skips animator assets to load more quickly. This helps diagnose runs that crashed, which sometimes causes the animator part of VS Visualizer to also crash. Also, the load time for VS files has been reduced dramatically, especially for larger files.

6. When VS Visualizer is showing plots with overlays from two or more output files, dragging new variables from the Data Manager window into a plot window adds the variables from all open files that contain those variables.
7. The default plot format was changed to match the style set in the **Plot Format** datasets shipped in recent versions of SuspensionSim, providing a more consistent appearance covering plots set up from the database and plots created interactively.
8. Additional keywords have been added to VS Visualizer to adjust the environment lighting. `CAM_GLOBAL_DIFFUSE` and `CAM_GLOBAL_SPECULAR` will adjust the lighting properties of the primary light source.
9. The Ambient Light setting has been removed from the VS Visualizer preferences and replaced with a Brightness Modifier. The Brightness Modifier is a numeric field where 1 indicates true light settings, less than 1 will make the scene darker, and greater than 1 will make the scene brighter. Any ambient light adjustments can be made from within the VS Browser using the keyword `CAM_GLOBAL_AMBIENT`. For examples, use the Find Text option (**Tools** > **Find Text in the Database)** and search for `CAM_GLOBAL_AMBIENT.`

New Examples

Two new suspension configurations have been added: a high-performance strut with separate steering knuckle, and a strut suspension with double ball joint. Three examples can be found in the category **Detailed Front Strut w Knuckle**, and three in the category **Detailed Front Double Balljoint Strut.**

SuspensionSim 2017.0 New Features

VehicleSim Architecture

Output Files

The SuspensionSim Solver can directly write comma-separated variable (CSV) text files as output options in addition to the existing ERD and VS binary outputs.

Commands originally introduced in SuspensionSim `LOAD_ERD_FILE` and `ERD_TO_TABLE` are renamed as `LOAD_TABLE_FILE` and `FILE_TO_TABLE` to data from an ERD or VS file to be copied into a table for use with a Configurable Function. Both of these are also available in CarSim and TruckSim as VS Commands, along with `FILE_TO_CARPET` and `ECHO_LOADED_TABLE`.

License Manager Changes

The stand-alone license manager, VSLM.exe, has been separated into separate products for BikeSim, CarSim, TruckSim, and SuspensionSim. The executables are named BSLM.exe, CSLM.exe, TSLM.exe, and SSLM.exe, respectively.

Licensing Interface Changes

Users will notice some changes to licensing behavior with version 2017.0. Only the first VehicleSim Browser window opened will ask for license selections. Additional windows will not show the licensing dialog. If the original window closes, then a single remaining Browser window will show the license dialog, and that particular window will be in control of the license settings for your machine. To suppress licensing messages over long sessions, run the stand-alone license manager (BSLM.exe, CSLM.exe, TSLM.exe, or SSLM.exe).

Graphical User Interface (GUI) and Database

Visual Appearance

The sizing of text and windows has been redone to work with Windows OS settings for display.

1. The font used throughout the GUI is now Segoe UI, which is the standard font used in the US versions of Windows. This font is also available in international versions. Most screens and popup dialogs have been redone such that displays are easier to read regardless of the Windows preferences.
2. The sizing options for all GUI windows are now tied to the Windows display preference. Text shown in menus and the window title always matches the OS preference (e.g., 100%, 150%, 300%, etc.). An option has been added to the legacy Viewing choices to also use the Windows display preference to size GUI windows. This provides full compatibility with newer high-definition monitors and laptops with small pixels.

New Results Section in Database

A new `Results` folder in the database provides the location for all output files generated when running simulations. This provides several advantages:

1. Output file names can be given short default names that are easily exchanged and manipulated by external software.
2. The `Runs` folder now contains only database files (Parsfiles and some image files).
3. The new organization is well suited to support the new feature to make multiple simulations from a single **Run Control** dataset as described in the next subsection.

Multiple Simulations from a Single Run Control Dataset

The **Run Control** screen has been extended to support the reuse of a single **Run Control** dataset to make multiple simulations without overriding previous results. This feature greatly extends the options for users running the models from external programs such as Simulink, especially when conditions are varied outside the scope of the CarSim database. For example, multiple tests made with HIL systems and driving simulators.

Output Files can be Included in CPAR

There is now an option to bundle run output files when creating a CPAR from the **Run Control** screen, or when adding runs from the **Library Tool**. This way a user importing the CPAR can immediately view results, even if they were made with advanced licenses not available for the person importing the data.

Cloning Options added to Dataset Rename

Options to clone an existing dataset, as well as re-routing links to the new copy, and deleting the old original, have been added to the **Edit > Change Title or Category of This Dataset** command (**CTRL + T**).

Updated Screens

The **Plot Setup** and **Plot Format** screens were updated to match new capabilities of VS Visualizer.

VS Visualizer

CSV Input

VS Visualizer can now handle CSV files for making plots and generating animations. This supports the new capability of VS Solvers for generating CSV files as another alternative to ERD and VS files.

Be aware that less labeling information is available for plots based on CSV files. The convention for spreadsheets is to have the first row provide names of the variables, with the rest of each column being a time history of the named variable.

Plotting

A number of improvements were made for the plotting within VS Visualizer:

1. Commands were added to manage the time used in zoomed plots and in the time control for the camera(s). Type ‘t’ to pan the active plot to the time used by the camera; type Shift-T to change the camera time to match the active plot.
2. The labels used to identify datasets in the legend are handled automatically to account for titles, categories, and file names. These support the new option for making multiple simulations from a single Run Control dataset.
3. Support was added for options available in the legacy WinEP program such as showing a frame around the plot area, not showing axes, and providing a fine grid.

Documentation Update

Minor edits were made throughout the Help documents. Documents that were changed to better describe the new version are listed here.

The following Reference Manuals were updated:

1. SuspensionSim Commands and Parameters Reference Manual
2. VS API
3. VS Commands
4. VS COM Interface
5. VS Solver Programs
6. VS Visualizer

The following Screen documents were updated:

1. Animator Reference Frames
2. Plot Screens
3. Procedures and Events
4. Tools > Calculator Tool for Tables
5. Tools > Batch Runs
6. Tools > Preferences

Bug Fixes and Errata

In versions prior to 9.0 (2014), 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.

Backward Compatibility

SuspensionSim 2017.0 will automatically convert databases going back to 8.2 (2015); this conversion takes care of the screen parsfiles and keywords such that the GUI will recognize old keywords and update them appropriately.

WinEP Plotter is Retired

The legacy plotter WinEP is no longer supported, in order to provide full support for VS Visualizer. GUI controls for workarounds needed for the legacy tool were identified in past versions with red text, announcing they would be removed in 2017.0. They are removed in 2017.0.

WinEP supported a moving average filter option, with a **Plot Transform (Legacy)** screen for settings. It was never used in SuspensionSim examples and has been removed.