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

SuspensionSim 2022.0 New Features

The last release of SuspensionSim was version 2021.0. This document lists the notable new features in SuspensionSim version 2022.0, organized into sections based on the main components of the software package.

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.

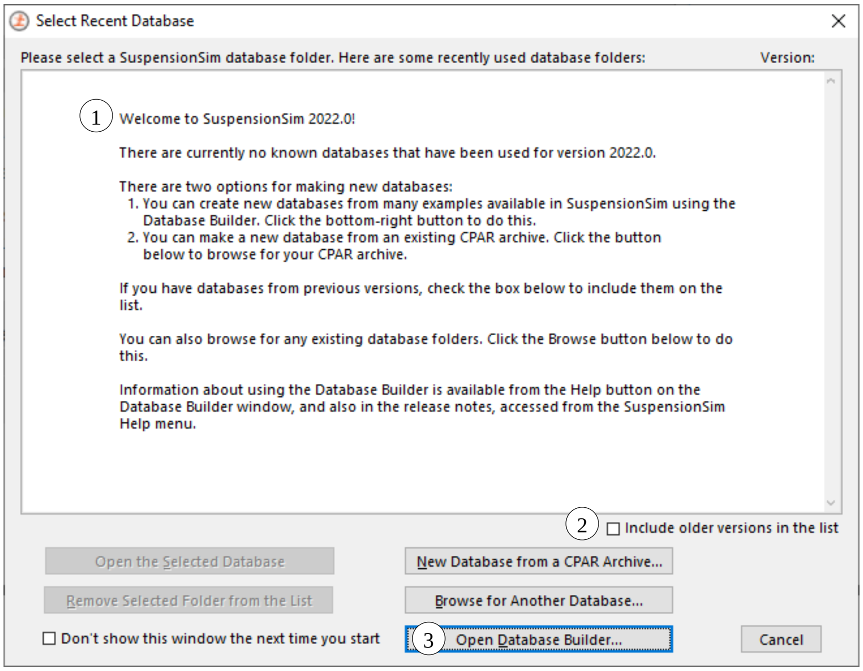
The new tools for installing databases are described in the following section. More information about the specific new datasets for this release is provided in the last section (Database, p. 10).

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 in the last section of these release notes, p. 10. For descriptions of all CPAR archives, please see the Help document for the Database Builder.

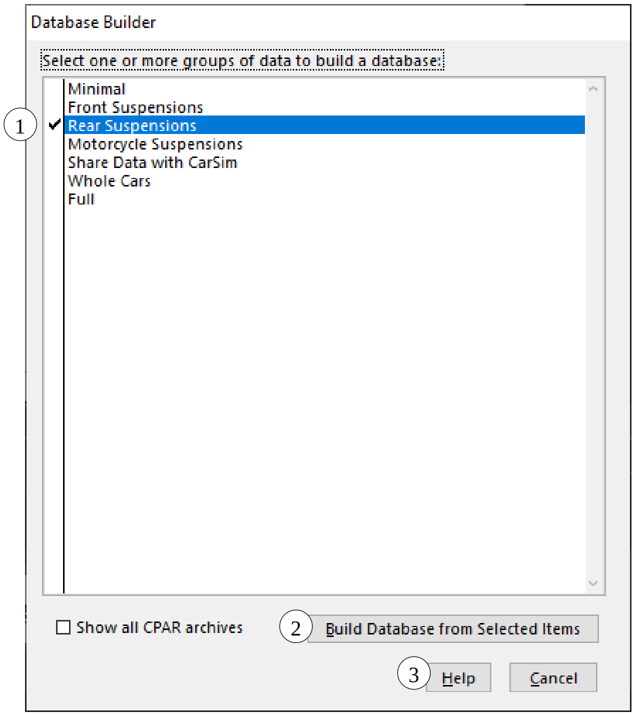


Figure 2. Database Builder window showing only predefined database options.

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
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

Table 1. Pre-defined database options.

Database Option	Description	Target Users
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.

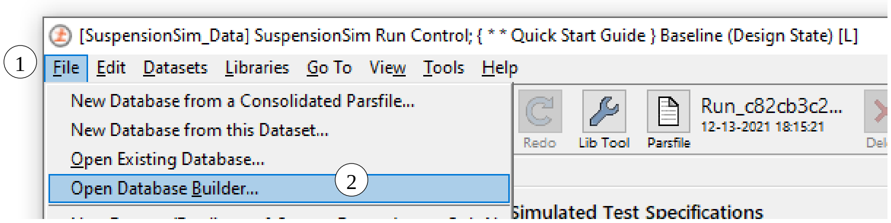
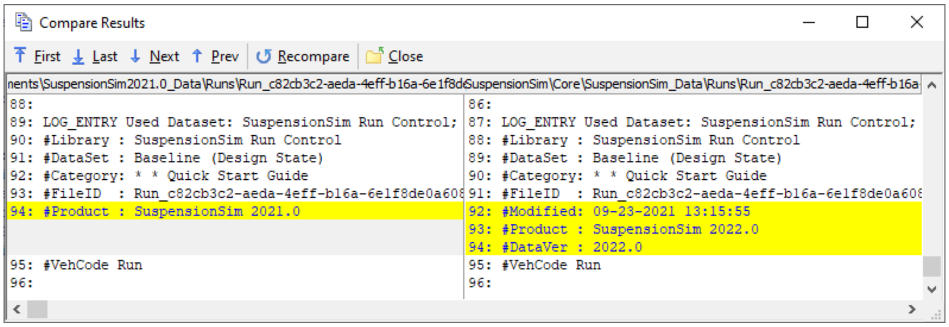

[SuspensionSim_Data] SuspensionSim Run Control; { * * Quick Start Guide } Baseline (Design State) [L]
1
File
Edit
Datasets
Libraries
Go To
View
Tools
Help
New Database from a Consolidated Parsfile...
New Database from this Dataset...
Open Existing Database...
Open Database Builder...
2
Redo
Lib Tool
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`.



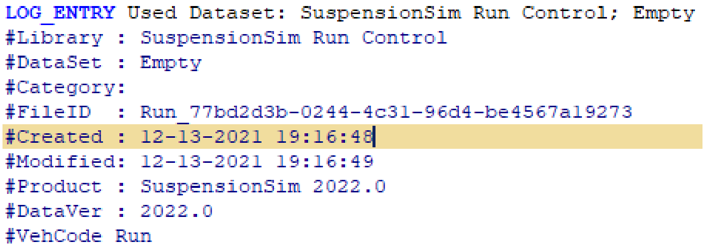
```
nents\SuspensionSim2021.0_Data\Runs\Run_c82cb3c2-aeda-4eff-b16a-6e1f8d  SuspensionSim\Core\SuspensionSim_Data\Runs\Run_c82cb3c2-aeda-4eff-b16a
88:                                                      86:
89: LOG_ENTRY Used Dataset: SuspensionSim Run Control;   87: LOG_ENTRY Used Dataset: SuspensionSim 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.

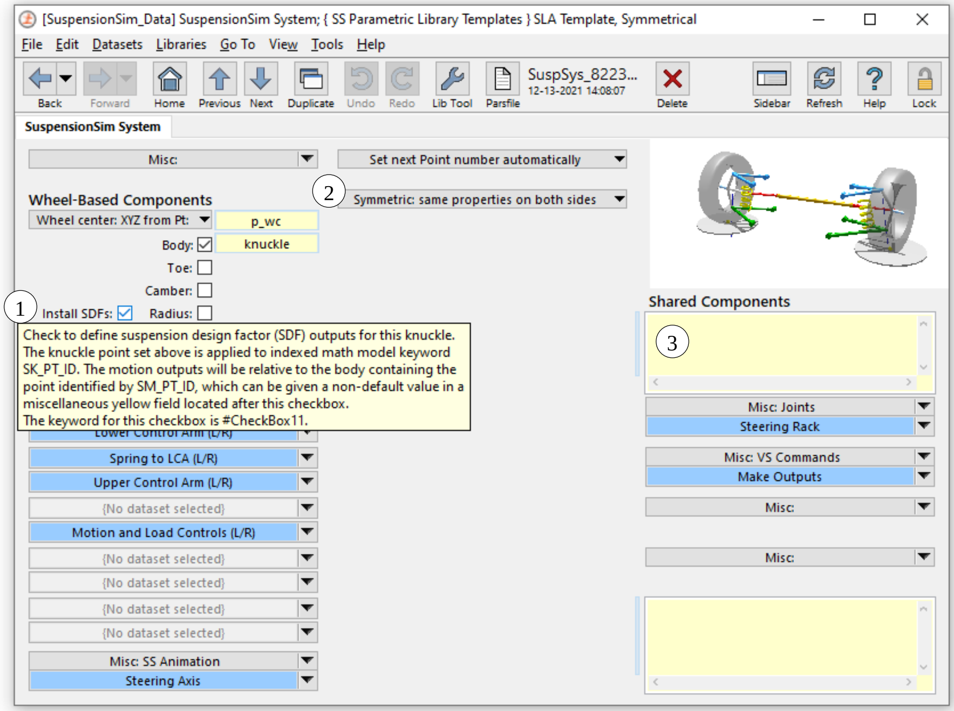
The new search option takes advantage of `Run_all.par` files that are made automatically whenever a run is made, or VS Visualizer is used to view results. This has significant advantages:

1. Searching these files is rapid compared to searching the entire database. Even with large databases, the search is much less than a minute.
2. The searching finds all references where the current dataset was used in an existing run.
3. 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.



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. 5, 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:

1. Animator: Reference Frame
2. Animator Shapes and Groups
3. Batch Matrix

4. Calculator: Symbolic
5. Generic Data Screens
6. Generic Table Screen
7. Setting up Import and Output Variables
8. External Models and RT Systems
9. VS Browser Preferences
10. Run Control Screen (Home)
11. SuspensionSim Model Screens
12. Writing Custom Parsfiles from SuspensionSim

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

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

13. SuspensionSim Commands and Parameters Reference Manual
14. System Parameters in VS Math Models
15. The VehicleSim API: Running and Extending VS Math Models
16. VS Browser Reference Manual
17. VS COM Interface: API of the VS Browser
18. VS Commands Reference Manual
19. VS Commands Summary
20. VS Math Models Reference Manual
21. VS Visualizer Reference Manual

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

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

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

28. Terms and Conditions of Sale and EULA
29. Network Licenses for VehicleSim Products
30. Silent Installation for VehicleSim Products

31. SuspensionSim Backward Compatibility
32. SuspensionSim 2022.0: New Features
33. VehicleSim System Requirements & Software Compatibility
34. Technical Support

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

35. 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 Suspension Design Factors, p. 6). 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 Database Builder, p. 2 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:

1. SS Library > Car
2. SS Library > Truck.

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

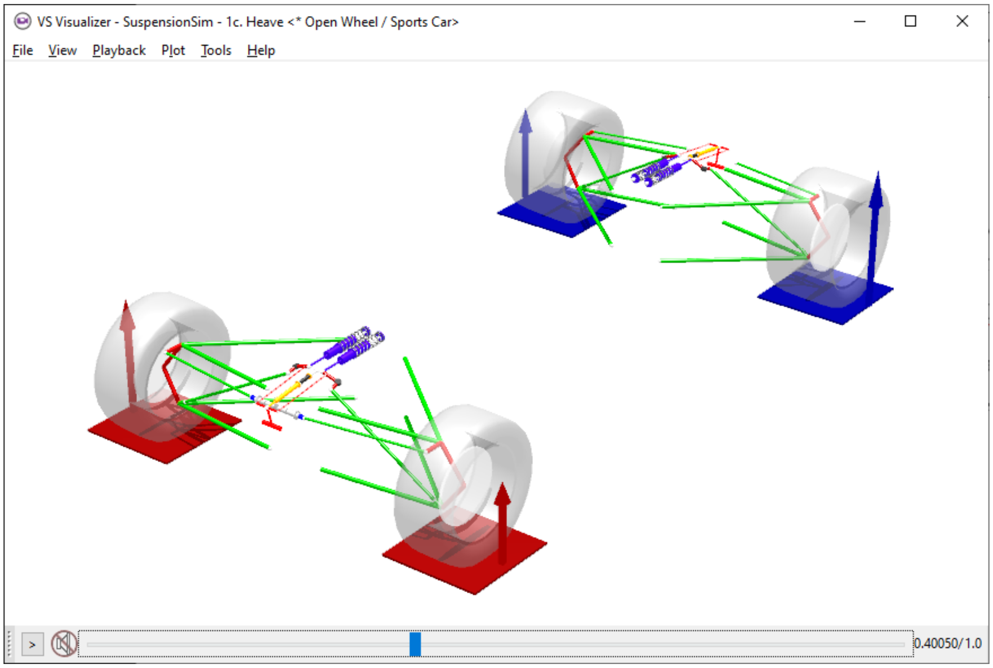

VS Visualizer - SuspensionSim - 1c. Heave <* Open Wheel / Sports Car>
File View Playback Plot Tools Help
0.40050/1.0

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

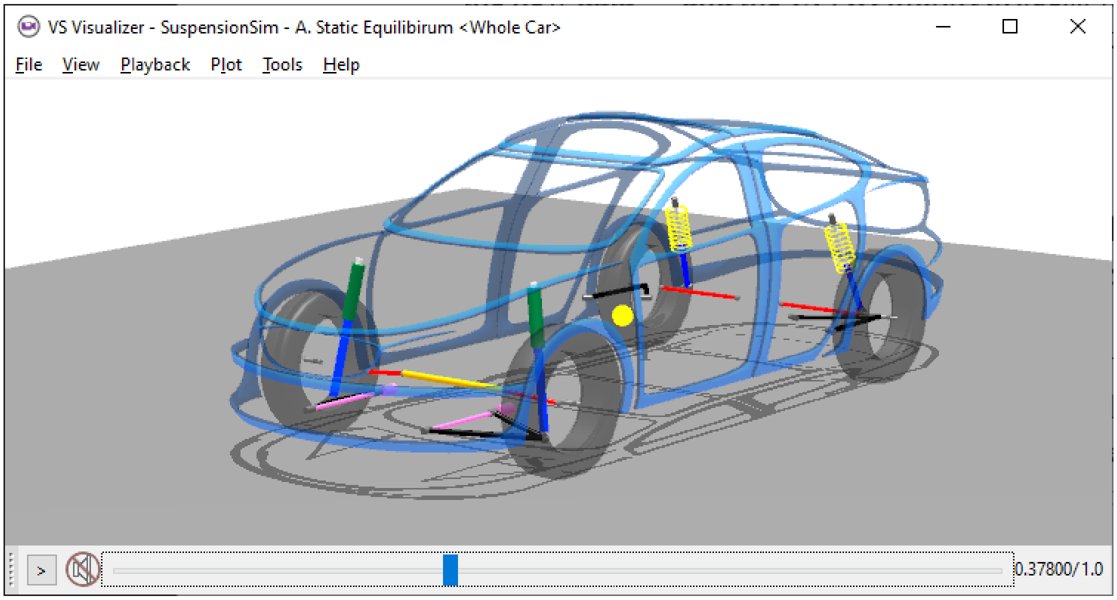

VS Visualizer - SuspensionSim - A. Static Equilibirum <Whole Car>
File View Playback Plot Tools Help
0.37800/1.0

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

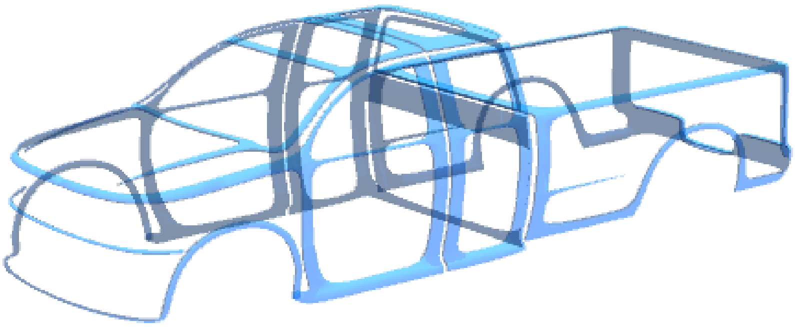


Figure 9. The new truck shape.