

# SuspensionSim 2016 Release Notes

- VehicleSim Architecture .....1
  - Echo Files .....1
  - User-Defined Names.....2
  - Output Variable Labels .....2
  - VS Commands .....2
  - Applying VS Commands and Statements with an API function .....3
  - Linking to VS Output files from BikeSim, CarSim, and TruckSim .....3
- Graphical User Interface (GUI).....3
  - VS/ERD File Utility.....4
  - I/O Channel Library Screens .....4
  - Miscellaneous Browser and GUI Improvements .....4
- VS Visualizer .....5
  - VSRAP File Support for Viewing Results Without SuspensionSim.....5
  - Copying Data Channels to Other Software.....5
  - Plotter Improvements.....5
- Documentation Updates .....6
  - GUI Documentation.....6
  - Reference Manuals .....6
  - Tech Memos .....6
- Example Models.....7
- Bug Fixes and Errata .....7

Starting with this release, Mechanical Simulation is following a calendar year convention for numbering versions for BikeSim, SuspensionSim, SuspensionSim, and TruckSim.

## VehicleSim Architecture

The mathematical modeling and calculation methods in SuspensionSim are mature and have not been modified. This release adds support for more documentation and information about user-defined parameters. It improves the options for moving data to and from SuspensionSim.

### Echo Files

Every VS Solver creates an Echo file that lists all parameters used in the run, along with the values assigned to the parameters. The main purpose of the Echo file is to describe each parameter in the model, with a keyword, value, units, and description. The information in the Echo file has been extended in SuspensionSim 2016 to customize the information written into the file to use names set by the user from the GUI.

The organization of the Echo file has been extended in several ways:

1. The file is organized by many more sections, where each section lists all of the parts of each type. There are new independent sections for Bodies, Points, Joints, and Vectors.
2. The file ends with a summary of all Bodies, Points, and Joints, described using the user-defined names.
3. Each part has a user-defined name to make the information more meaningful with respect to a specified model. For example, here is part of the summary the lists the Points:

```
! 14 Tie rod, L, JP, located in the Steering rack
! 15 Tie rod, L, CP, located in the Knuckle
! 16 LCA ball joint, L, JP, located in the Knuckle
! 17 LCA ball joint, L, CP, located in the Lower arm
```

And here is a part of the listing for Joints:

```
! 10 Front UCA bushing, L is a Bushing between the Ground and the Upper arm using
!   Points 28 and 29

! 11 Rear UCA bushing, L is a Bushing between the Ground and the Upper arm using
!   Points 31 and 32

! 12 Control Z of WC, L is a 3D connection between the Ground and the Knuckle
!   using Points 34 and 35
```

## User-Defined Names

A SuspensionSim model has a relatively small set of parameter names that are indexed for the Bodies, Points, Joints, and Vectors that are assembled to define the multibody model.

SS Solver includes a parameter for each part that exists solely to give a name to the part. The names are written into the Echo file. The names are also used during initialization to provide custom descriptions for other parameters that are related to the part.

## Output Variable Labels

Each Body, Point, Joint, and Vector included in the model causes SS Solver to create a number of output variables that are updated for each calculated equilibrium condition in a simulation. The number of output variables may range from hundreds to thousands, depending on the complexity of the model. Each variable is identified with a short name with a rigid naming convention involving the type of variable (X coordinate, Y coordinate, Force, Moment, etc.) and part number. Each output also has several longer and more descriptive names that are used for labeling plots. These labels are generated from the custom names provided in the GUI datasets, greatly simplifying the work needed to make plots for reports.

## VS Commands

Several improvements were made in the handling of VS Commands and VS assignment statements that occur when scanning inputs from a Parsfile.

1. Assignment statements and VS Commands that define new variables have been extended to include a description for the variable being defined or assigned. In past versions, the

variable could be assigned using a number or formula, and the units could optionally be specified if the statement included a semicolon delimiter. These commands have been extended to include descriptive text, specified if a second semicolon is included in the statement.

2. All self-contained lines can be split using a continuation delimiter of three dots '...' as is done in MATLAB. This is helpful for VS Commands with long formulas, or any lines in which alternate text descriptions are provided for installed variables.
3. Equations added with VS Commands such as EQ\_IN can be used to assign new values to integer parameters. In such cases, the floating-point value from the formula is rounded to the nearest integer value. (In past versions, equations from VS Commands such as EQ\_IN could only be used to set new values for floating-point parameters and variables.)
4. The error handling of VS Commands has been made more aggressive, such that a diagnostic error message is always produced if a command is recognized but has missing information.

### **Applying VS Commands and Statements with an API function**

A new API function was added: `vs_statement`. This has just two string arguments: an uppercase keyword and the text that would follow the keyword in a Parsfile. This will then be processed using the same internal input handling that is applied for lines of text read from a Parsfile. It should handle any self-contained statement that can currently be provided in a Parsfile. (The only type of data not supported is tabular data for a Configurable Function, which requires multiple lines of text.)

This adds support for options such as setting up Import and Export variables programmatically, rather than through the Parsfile.

A new location was also added for callback functions in support of the use of the `vs_command` function during the same stage of the simulation setup as when the Parsfile is read; this is just after the VS Solver has finished reading the Parsfile but before the internal initialization starts.

### **Linking to VS Output files from BikeSim, CarSim, and TruckSim**

SuspensionSim includes commands to obtain data from simulation output files from VS Math Models in CarSim, TruckSim, or BikeSim, for study of movements and Joint forces and moments. This capability has been extended from the original ERD files to also work with the new VS file format. Besides supporting 64-bit binary data, VS files allow longer names for variables and labels, which will be used in future versions to better handle long text names for variables.

### **Graphical User Interface (GUI)**

The SuspensionSim browser program `suspensionsim.exe` includes improvements to some of the screens.

## VS/ERD File Utility

SuspensionSim has included an interactive tool used to convert the output files generated by VS Solvers to formats that can be used in Excel and MATLAB. It also supports the conversion of Excel files to ERD format for use with VS tools.

The old tool did not support the new VS file format introduced with Version 9. The tool has been updated to support the VS file formats (32-bit and 64-bit) and provide better browsing of files using the new longer machine-generated names in the SuspensionSim database.

## I/O Channel Library Screens

SuspensionSim includes two library screens where lists are specified for Output variables of interest; these are the **I/O Channels Write** and the **Plot Setup** library screens.

Both of these library screens include browser controls for selecting variables to Plot or Write to File by pointing and clicking.

The process of refreshing the screen display for each specific suspension was improved and made automatic, so the browse information is always shown instantly when viewing a dataset from any of these libraries when a reference **Run Control** dataset was selected.

The output variables have multiple labels for each variable. The displays were improved to always show both the short unique name along with a longer, more descriptive name, as well as the user-units of the selected variable. The names are sorted alphabetically by the short names, which are created using consistent conventions (X coordinates begin with 'X', Forces begin with 'F', etc.).

The screen for the **I/O Channels: Write** library now provides full support for automatic generation of output files formatted for use in Excel (CSV files) and MATLAB (binary MAT files). It also allows for adding a filename suffix if desired.

## Miscellaneous Browser and GUI Improvements

1. The option to search for text in the database (**Tools > Find Text in the Database**) now has the keyboard shortcut Ctrl+F, and supports two more options: **Find whole words only** and **Current Library only**. The new options are in addition to the previously available options **Allow wildcard** and **Case sensitive**.
2. The SuspensionSim modeling screens (**Locations, Joints, and System**) were updated to generate better labels and names, making use of new features of the VehicleSim architecture.
3. A checkbox was added to the **SuspensionSim Location and Bodies** screen to automatically generate labels for Bodies on the right side of a suspension using names of Bodies on the left side.
4. Blue links were added to the **Animator: Reference Frame** screen to allow single animator shapes to be specified on the same screen. This allows a single dataset to be used to view or animate simple objects.

5. Screens that set values for legacy features in the WinEP or SurfAnim programs that are not relevant for VS Visualizer identify those features with red text. They will be removed in 2017.
6. Low-level changes were made to speed up the launching of the program.
7. Improvements were made for controlling the import of CPAR archive files from one database to another.

## VS Visualizer

VS Visualizer adds features to share data and improve the plotter usability.

### VSRAP File Support for Viewing Results Without SuspensionSim

A VehicleSim Results Analysis Package (VSRAP) is a file that contains all of the necessary data to animate and/or plot the results of one or more simulation runs. VS Visualizer now supports both creating and viewing VSRAP files. A VSRAP file can be created by VS Visualizer on one computer, and then transferred to another computer where it can be opened with VS Visualizer. The second computer does not need to have SuspensionSim installed, but does have to have the VS Visualizer installed.

An installer for VS Visualizer as a stand-alone application is available from [www.carsim.com](http://www.carsim.com).

### Copying Data Channels to Other Software

Data Channels can now be copied to other software using Drag-and-Drop or the Windows system clipboard. To use this feature, select one or more Data Channels in the Data Manager window and either drag them to the other software package, or press Ctrl-C to copy them to the system clipboard. The Data Channels are copied in CSV text format. The Data Channels can be copied to a text editor, spreadsheet, MATLAB, etc.

Overall, there are now three methods for getting simulation outputs into Excel or MATLAB:

1. Drag them or copy them from VS Visualizer (good for occasional transfers).
2. Use the VS/ERD File Utility to convert a complete VS or ERD file (good to get occasional files or files generated from past work).
3. Link to an **I/O Channels: Write** dataset when making new runs, to generate the Excel or MATLAB files as each new simulation is run (easiest for new runs).

### Plotter Improvements

1. Font type, size, color can be specified for Legend, Axis Labels, Tick Numbers.
2. A Fine grid is now supported in addition to course grid.
3. Improved interactive mouse control: middle mouse button and middle mouse button emulation improved.
4. Added ability to copy a bitmap image of the plot to the Windows clipboard, and/or save the plot as a bitmap file.

5. Support auto scaling and manual scaling on each axis independently (e.g. automatically scale the X axis, but specify range for Y).

## Documentation Updates

SuspensionSim includes about 1000 pages of PDF documentation, accessed through the **Help** menu, the **Search Help** option, and using the F1 key for any screen.

### GUI Documentation

1. The document **SuspensionSim Models** was updated.
2. The document **Write VS Parsfiles** was updated.
3. The User Manual for the ERD Converter was completely replaced with new documentation for the replacement **VS/ERD File Utility** tool.
4. The document **Setting up Import and Output Variables** was updated to describe the new library GUI for I/O screens, with automatic browsing of Output variables, and the new interface to generate Excel and MATLAB files.

### Reference Manuals

5. The **SuspensionSim Commands and Parameters** Reference Manual was revised to describe the automatic customization of parameter descriptions and output variable labels. Information about the GUI and VS Visualizer was moved to other documents.
6. The **VS Solver Programs** Reference Manual was updated to include the new three-dot continuation option supported for reading text from Parsfiles and the new in-line format for setting value, units, and description for parameters.
7. The **VS Commands** Reference Manual was updated to add options for setting descriptions for new and existing variables, and details on setting units for Configurable Functions.
8. The **VS API** Reference Manual was updated to add documentation for the `vs_statement` function and a new callback location `VS_EXT_AFTER_READ`. The organization was updated to have more information about steps in the simulation in Chapter 2, and more information about MATLAB usage.
9. The **VS COM Interface: API of the VS Browser** manual was updated to include new commands.

### Tech Memos

10. The tech memo **Guidelines for Building SuspensionSim Models** was updated.
11. The tech memo **Reusable Datasets in SuspensionSim** was updated.
12. The tech memo **Automating Runs with the VS API** was updated to show newer examples.

13. The tech memo **Example: Extending a Model with VS Commands and API** was rewritten to give more information about MATLAB and to use the new `vs_statement` API function.

## Example Models

All of the examples from Version 9 (2015) are still present in 2016. Improvements in the labeling for the Echo file and the output names written for plotting and animation are applied automatically when an old database is re-written for version 2016. In addition, most of the existing datasets were modified to provide better documentation:

1. All variables (parameters and outputs) added with VS Commands now include full descriptions that are written to the Echo file.
2. The two detailed examples now use Symbol Stack variables to remove hard-coded references to Point and Joint numbers, making the datasets easier to modify and maintain.
3. The recently introduced `SYMBOL_SET_GLOBAL` command is used in some examples to share information between low-level and high-level datasets.

## 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. Symbol stack management was improved, and behavior of the `SYMBOL_SET_GLOBAL` command was modified to work more easily.