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 2020.0 Release Notes

The last release of SuspensionSim was version 2019.0. This document lists the notable new features in SuspensionSim version 2020.0, organized into sections based on the main components of the software package.

The SuspensionSim Installer

After the initial installation of SuspensionSim, the installer

`Setup_SuspensionSim_2020.0_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, 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.

Model Features

VS Commands

The VS Commands Reference Manual has been updated.

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