Mechanical Simulation

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

BikeSim 2018.0 Release Notes

Architecture Changes

Longer Names for Output Variables

A new VS output file format was introduced in 2014 as an alternative to the legacy ERD file format. In version 2017, 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 now make use of longer names in order to provide consistent naming of variables when there are many instances (e.g., more than nine ADAS sensors). If the output variable has a name longer than 8 characters, both VS and ERD output files include both the long name and a name limited to 8 characters. VS Visualizer and 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 will still be processed correctly.

If the ERD format is used, the header now includes both long and legacy names. The keyword `SHORTNAM` has been used since the ERD format was published in 1984. VS Solvers from version 2018 provide these names, and names that may be longer, using a new keyword `NAME`. Because the `SHORTNAM` information is still written to the ERD header, legacy 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 versions for 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**.

Binary Table Files

Tabular data used in Configurable Functions have traditionally been provided in text Parsfiles. In some cases, very large tables are needed to represent road surface properties. A result is that the time needed by the VS Solver to read the input Parsfiles can be excessive. For example, some datasets used in driving simulators might take more than 20 minutes to load.

Version 2017 included some example simulations that took place in Mcity, with the ground geometry provided by 2D tabular data read from an ERD binary file. In this case, the time to load the binary data is about 0.15s (reduced from about 20 minutes for a text version).

In version 2018, An API (`vs_table.dll`) has been developed to support this type of usage, with utilities for creating binary table files (`.vstb`) that are well-suited for reading large tables in a short time.

An API (`vs_output.dll`) has also been developed to support the reading and modification of VS Output files (`.vs`) that will aid the user in modification of VS Outputs for use by other programs or processes.

Full Support for User-Defined Differential Equations

VS Commands were introduced in BikeSim in 2007, including the option to add ordinary differential equations (ODEs) with the command `EQ_DIFFERENTIAL`. The ODEs added with VS Commands have been integrated with the same methods as the built-in ODEs. However, some features were not fully supported.

In version 2018 user-defined state variables are now listed in machine-generated documentation files (accessed with the View button on the lower-right corner of the Run Control screen), the same as built-in state variables.

User-defined state variables are now listed in the End Parsfile (along with built-in state variables) to support a new run that continues from the end of an existing simulation run.

Error and Range Checking in VS Command 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 2018, 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.

Writing Output Variables for Optional Parts

BikeSim has many output variables that are associated with the basic parts of the multibody model, such as sprung mass motions, tire kinematics and forces, suspension deflections and forces, etc. These variables always exist, and can be plotted, applied to animate auxiliary objects (e.g., force arrows), and used in VS Commands. The variables are activated for writing to file by referring to them in datasets from the **Plot Setup**, **Animator Reference Frame**, and **I/O Channel Output** libraries.

The math model also has many parts that are optional (e.g., reference points) and also have associated output variables. The output variables for the optional parts are added to the model when the parts are defined automatically by linking to datasets that define the properties of the parts.

The default behavior for VS Commands that create new output variables was changed in v2018 to define the variables, but not activate them for writing to file. When the checkbox “Write all outputs” on the **Run Control** screen is not checked, outputs are activated for being written to the output file only if they are used in animations or plots. Of course, they still exist, and may always be used in VS Commands and/or exported to other software.

In some ADAS scenarios, there can be tens of thousands of output variables associated with moving objects and detections by ADAS sensors, so this behavior could result in very large output files. The new behavior supports the creation of much smaller files, and sometimes faster simulation.

Model Features

Vehicle Dynamics

New Parameter for Three-Wheeled Motorcycles

A new parameter was added to the three-wheeled bike model (root keyword is `A_LEAN_LINK`). This is an angle for the rotation axis for the linkage in the steering columns. In previous versions, the axis was perpendicular to the steering column; in version 2018 the linkage can be parallel to the ground or any arbitrary angle.

Rolling Resistance

An import variable `IMP_RR_SURF` was added to adjust rolling resistance, to extend the native parameter (`RR_SURF`).

GPS Outputs

Output variables for GPS latitude and longitude in past versions were based on parameters for reference latitude and longitude for global X-Y coordinates of (0,0). Three new parameters were added for version 2018 in support of large changes in global Y coordinates. These are reference X and Y coordinates corresponding to the reference GPS latitude and longitude, and a limit of Y change that will trigger a new GPS calibration.

Paths and Roads

The maximum number of reference paths was increased from 100 to 200. The maximum number of supporting tables such as `LTARG` (lateral offset) and `SPEED_TARGET` (target speed for ego vehicle or moving object) was also increased to 200.

When working with external software that provides road surface information, there was previously a check to ensure that all information was provided. For each point of interest, the import variables are `Zgnd`, `Mux`, `Muy`, `Dzdx`, and `Dzdy`. The error-checking has been removed in support of setups where `Zgnd` is provided but other variables can keep internal values.

ADAS (Sensors and Moving Objects)

Many improvements have been made involving the ADAS sensors and moving objects.

Optional writing of output variables

In some ADAS scenarios, there can be tens of thousands of output variables, so this behavior could result in very large output files. As mentioned earlier, the default behavior was changed for version 2018; the outputs are not activated by default. When the checkbox “Write all outputs” on the **Run Control** screen is not checked, outputs are typically activated by using them in animations or plots. This means the output files for ADAS simulations in version 2018 may be significantly smaller, and the simulations can run faster.

Moving Objects

The maximum allowable number of moving objects was increased from 99 to 200.

More support has been added for using moving objects to represent traffic vehicles. A speed variable V_OBJ_*o* has been added to provide a consistent differential equation for calculating station. A parameter OPT_SPEED_OBJ was added with three options for handling speed: 0 -> don't use speed to calculate station or X-Y coordinates, 1-> integrate V_OBJ_*o* to obtain station, and 2-> integrate V_OBJ_*o* to obtain station and also use a new wheelbase parameter L_WB_OBJ to calculate heading angle with a differential equation that simulates vehicle offtracking. When offtracking is enabled, the motions of the moving objects resemble the behavior of real vehicles even in severe low-speed turns. When OPT_SPEED_OBJ is zero, the variable V_OBJ_*o* is calculated using finite differences.

X and Y coordinates for added for the four corners of rectangular objects. For cylindrical objects, the four points are for the front, right, rear, and left.

GPS latitude and longitude were added as output variables for all moving objects.

Information about an object shape is always used if there is at least one ADAS range and tracking sensor in the simulation. A parameter named OPT_SHAPE_OBJ was added to also use shape information for moving objects in case there are no ADAS sensors.

ADAS Sensors and Detections

The maximum allowable number of ADAS sensor was increased from 20 to 99.

A range-limit parameter RANGE_LIMIT_SENSOR was added to the sensor model. It is used to eliminate calculations for moving objects that are guaranteed to be out of range for a sensor; this can reduce the simulation time noticeably when there are many objects and/or sensors.

A similar method is used to eliminate moving objects based on bearing angle, using the existing field of view parameters.

Another parameter was added for sensors in general. When the new parameter OPT_ALTERNATE_SENSORS is given a non-zero value, only one sensor is updated each time step. The sensor being updated alternates among all sensors. For example, if there are four sensors, each one is updated every fourth time-step. In nearly all cases, the parameter can be safely given a nonzero value; the default value is 1. When the value of this parameter is zero, all sensor detections are calculated every time step, as was done in past versions.

The new parameters RANGE_LIMIT_SENSOR and OPT_ALTERNATE_SENSORS can speed up the simulation, with the difference being more significant for large numbers of moving objects and sensors. For a test case with 20 sensors and 99 moving objects, the calculation decreased by a factor of 10 (90% reduction in computation time)!

The connection to the main VS Solver was redone for 2018 to provide better support for VS Commands and Events. Past versions did not support changes to path or road ID numbers after the run initialized; version 2018 provides fully control for changing properties with events. In past versions, initialization was not complete at T=0; now it is.

Output variables for detections all involve the connection between one sensor and one moving object. Each time a sensor or object is created, output variables for all potential detections are created. In past versions, output variables for detections could not be used in VS Command equations until both the sensors and moving objects had been defined. BikeSim 2018 adds a new sensor parameter N_MIN_DETECTIONS that causes the DEFINE_SENSORS command to create a minimum number of sets of detection output variables. If left at the default value of 1, each time a sensor is defined, the output variables for one detection will also be created. This feature enables sensors to be created before moving objects, as is the case in the BikeSim GUI if the sensor datasets are linked to a vehicle dataset.

New detection output variables were added: the global X and Y coordinates for the left and right edges of a detected object.

A new algorithm was implemented to control occlusion of objects. In version 2018, a parameter for each sensor OPT_LEGACY_SENSOR can be set to 1 to disable the new algorithm (the default is 0). This new algorithm provides increased accuracy in certain edge cases.

VS Visualizer

The VS Visualizer Application

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 (or other video players).

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 used for plots of trajectories of points on the vehicle and/or moving objects, such as offtracking plots.

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 cause the animator part of VS Visualizer to also crash. Also, the load time for VS files has been reduced dramatically, especially for larger files.

When VS Visualizer is shown 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.

The default plot format was changed to match the style set in the **Plot Format** datasets shipped in recent versions of BikeSim, providing a more consistence appearance covering plots set up from the database and plots created interactively.

Animation Resources

The amount of animator resources provided in BikeSim has been greatly expanded. All current vehicle shapes for CarSim, TruckSim, and BikeSim are now provided in all three products, in support of ADAS examples with a variety of traffic vehicles. The Resources/Animator folder in BikeSim_Prog has animator assets organized into five top-level folders:

1. 2D_HUD_Icons
2. 3D_Shape_Files
3. Audio_All_VS
4. Road_Materials

5. `Deprecated_Assets`

Older versions of the software had many example shapes that used older formats as required by the legacy SurfAnim program. Shape files have been upgraded to take advantage of more modern options, such as compression and lighting/shadow support.

The folder `Deprecated_Assets` contains assets that were used in 2017 but are no longer used. It is provided to in order to support databases imported from older versions.

This release has other new features, such as alternative drivers (male and female, Caucasian and Asian) and more traffic signs.

New Tools

Version 2018 introduces new interactive tools for assembling scenes, paths, and road surfaces for ADAS examples. These include:

- **VS Scene Builder** uses drag-and-drop tiles to rapidly assemble scenes with 3D animation assets. The tool also provides click and drag options for creating VS reference paths to use with BikeSim close-loop controllers and moving objects. Access this tool from the BikeSim **Tools** menu.
- The Atlas web server (atlas.carsim.com) introduced in BikeSim 2016 was extended to the third generation to support the new import tool and to provide elevation data associated with downloaded paths.
- A new plug-in for the ADAS RP application leverages the HERE map database for detailed road information with an option to query elevation data using Google via the `carsim.com` Atlas web service. It combines horizontal coordinates from HERE with Z coordinates from Google to convert real-world map information into path and road datasets that can be used within BikeSim.

These new tools generate JSON (JavaScript Object Notation) text files that can be imported to BikeSim with the new **Scene: External Import** screen described below.

Graphic User Interface (GUI) and Database

Scene: External Import Screen

A new library **Scene: External Import** was added to support import of path, road, and animation datasets from Atlas, ADAS RP, and VS Scene Builder. Datasets from this library are linked to datasets in the **Road: 3D Surface** library.

The imported datasets are unique in having both path/road data for the VS Solver, along with supporting 3D animator assets.

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 and for shapes used to show targets in ADAS applications. These links led to problems in some cases when transferring data with CPAR files, so they were removed for versions 2018. 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. Checkboxes for selecting the deprecated tools WinEP and SurfAnim were removed; BikeSim 2018 does not support those tools.

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. The blue links for default cube and cylinder used for ADAS-related animations have been removed. The link to 1x1x1 cube or cylinder is now hard-coded. (It still applies only when a box is checked to view the pure target shape.)

Moving Object Screens

The **Multiple Moving Objects** screen was changed to support new features. A wheelbase parameter and option were added in support of offtracking. The VS Commands that are written automatically make use of the new speed variable `V_OBJ_`*o*. `EQ_DIFFERENTIAL` commands that were used in past versions are no longer needed and are no longer written to the Parsfiles.

When the checkbox **Show target?** is checked, a rectangular or cylindrical shape is shown by VS Visualizer if a simulation using the dataset is viewed. In past versions, the 3D shape file was set in the **Preferences** screen. As noted in the previous subsection, the database now automatically uses a 1x1x1 cube or cylinder shape.

The **Single Moving Object (Custom)** screen was also changed to support the new speed variables and remove some automatically written VS Command equations.

Miscellaneous Changes

Here are some other changes to the GUI.

- Many of the miscellaneous yellow fields are now resizable to provide a quick view of the contents without scrolling.

- The **Tools** menu adds support for MATLAB 2017b.

- 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.

New Examples

1. BS 2018 – New Animator Assets. A few runs were made that show some of the new animator assets added to BikeSim.
2. BS 2018 – Scene Import. New datasets exercising the new Path import method show scenes, paths, and roads imported from ADAS RP, VS Scene Builder, and Atlas.
3. BS 2018 - Three-wheel linkage angle. This example shows the new capability to define the linkage angle on the front suspension of a three-wheeled motorcycle.

Documentation Updates

Documents from the Help Menu

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

1. ADAS RP Plug-In (Tech Memo)
2. Scene External Import (Screen Documentation)
3. Validation of VS Vehicle Models (Tech Memo)
4. VS Scene Builder (Tools submenu)
5. 3DS Max OBS OSG Export Setup (Tools submenu)
6. VS Output/Table API (Reference Manual)

The following Reference Manuals have been updated:

7. System Parameters in VS Solvers
8. VS API
9. VS COM Interface
10. VS Solver Programs
11. VS Visualizer

The following Screen documents have been updated:

12. ADAS Sensors and Moving Objects
13. Animator: Camera Setup
14. Animator Shapes and Groups
15. Animator Sounds
16. Animator: Wheel Arrows and Other Indicators
17. Custom Forces and Motion Sensors
18. External Models and RT Systems
19. Plot Screens

20. Preferences
21. Run Control Screen (Home)
22. Setting Up Import and Output Variables
23. Three-wheeled motorcycles
24. Tire Models

The following technical memos have been updated:

25. Example: Camera Sensors
26. Example: Running Multiple Vehicles in Simulink
27. Making Advanced Procedures with VS Commands
28. Numerical Integration in VS Solvers
29. Vehicle Dynamics Terminology

The following documents involving RT systems have been updated:

30. Concurrent RT Guide
31. dSPACE RT Guide
32. RT Lab Guide
33. NI RT Guide

Machine-generated documents

The machine-generated documentation for output variables includes more details for some variables, such as coordinate systems, details of definitions, etc. If the name of the variable is more than 8 characters in length, a shorter name is also listed that can also be used.

The machine-generated documentation for state variables now include ODE state variables that were created with the `EQ_DIFFERENTIAL` command.

Values of state variables created with the `EQ_DIFFERENTIAL` command are now listed in the End file, in support of continuation runs.

The math models within the VS Solvers include some variables that were introduced in machine-generated code as independent state variables that are then constrained to be algebraic functions of other state variables. Past versions listed the constrained state variables in machine-generated documentation (accessed with the **View** button in the lower-right corner of the **Run Control** screen), with the misleading implication that they could be given initial condition values. In version 2018 they are not listed.

Bug Fixes and Errata

Miscellaneous

1. The occlusion of moving objects that are detected by ADAS sensors was based on sorting the objects based on distance from the sensor. The simple method used in past version did not always work as intended in some cases.
2. Tire forces and moments at the center of tire contact (CTC) may be imported. However, if the selected tire model was an external model such as TASS MF-Tyre or COSIN/Ftire, the import variables were not used in the multibody model, but were added to the force and moment output variables. This was misleading. In version 2018, these imports are ignored completely for external tire models (`OPT_TIRE_MODEL` ≥ 5) and do not produce confusing outputs.
3. Texture render issues which occurred when viewing Mcity scenes in VS Visualizer's compatible mode have been corrected.
4. The api-ms-win-crt-runtime*.dll error when using Windows 7 SP1 has been addressed with v2018.0.
5. Run and CreateAllParfile COM functions failure on locked screen has been fixed. User may have to unlock a screen before executing COM functions which would modify screen data.
6. Compatibility problems with MATLAB 2017a and 2017b have been corrected.

Backward Compatibility

Miscellaneous

1. Simfiles must have a vehicle code in order for the VS Solver to activate the appropriate modules. This is done automatically when runs are made from BikeSim GUI, and it must also be done when Simfiles are used by external software.
2. The default behavior for VS Commands that create new output variables was changed in v2018 to define the variables, but not activate them for writing to file. (This is the same behavior as all built-in output variables.) Variables are activated for writing by using them in datasets from the **Plot Setup**, **Animator Reference Frame**, and **I/O Channel Output** libraries.

   Be aware that output variables cannot be activated until after they are defined. Therefore, datasets to create new parts (sensors, moving objects, etc.) must be referenced before the datasets that will make use of them.
3. Output variables used to support fake shadows for animations in the obsolete SurfAnim program have been removed from the solvers.

4. The solver now checks for exceptional floating-point values such as NaN at simulation time. When importing a database from an earlier version of BikeSim, some existing VS commands equations may need to be rewritten.

Known Issues

1. When working with Windows 10 and multiple monitors, and the monitors have different DPI settings, the menus of BikeSim 2018 do not position correctly on at least one of the monitors. Once a BikeSim menu is activated, the mouse position is offset or wrong until the menu is dismissed or an item is selected. To avoid this behavior, ensure that the BikeSim pop-up windows are positioned on the screen where you intend to work.

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