System Parameters in VS Math Models

Mechanical Simulation

Table of Contents

Last updated May 2022.

Introduction

Mechanical Simulation Corporation produces and distributes software tools for simulating and analyzing the dynamic behavior of motor vehicles in response to inputs from steering, braking, throttle, road, and aerodynamics. The simulation packages are organized into families of products named BikeSim®, CarSim®, and TruckSim®. Another product, SuspensionSim®, simulates quasi-static equilibrium conditions for suspensions and other multibody systems. All are based on the simulation architecture named VehicleSim®.

The Windows versions of all VehicleSim products include a main program that provides a GUI for managing a database of vehicle and test descriptions, running the simulation models, visualizing results through animation and plotting, and accessing documentation. The main programs for the products are named `bikesim.exe`, `carsim.exe`, `trucksim.exe`, and `suspensionsim.exe`. The VS Browser is only available on Windows; on Linux, the simulations are run using the command line tools. This manual applies for all products, and therefore the main program is called by the generic name *VS Browser*.

A *VS Solver* is a library in a VehicleSim product with functions that are used to automatically construct a *VS Math Model*, which then runs the simulation. Although VS Math Models for different products support different model features, all operate the same way when it comes to reading and writing files and communicating with other software. VS Math Models with differential equations (e.g., vehicle models in CarSim, TruckSim, and BikeSim) all use the same methods for solving the differential equations.

This manual provides details about parameters that can be used to customize the simulation in all VS Math Models, regardless of the details of the model. The main parameters are set in the GUI of the product. However, advanced users can set others by typing keywords and values into miscellaneous yellow fields available on many screens in the VS Browser.

This manual is a continuation of the material presented in the **Help** document **Reference Manuals > VS Math Models**, which describes the files that are read as input and written as output in the simulations.

Echo Files

When a VS Math Model runs, it creates two summary files that list every parameter and table that was used. One is created before the run, and another is created at the end of the run. For example, the files created for a simulation with a base name of `LastRun` would be `LastRun_echo.par` and `LastRun_end.par`. Both files contain all parameters that were used by the VS Math Model. As indicated by the `.par` extension, the `_echo` and `_end` files follow the Parsfile format used for all files used as inputs to a VS Math Model. Figure 1 shows the top of a CarSim Echo file with the name `LastRun_echo.par` (line 10).

An Echo file documents the options that were available within the VS Math Model used for the run; every parameter or table that influenced the simulation is shown in the Echo file.

The top of the file has system parameters that exist in all VS Math Models, such as the time step, start time, stop time, etc.

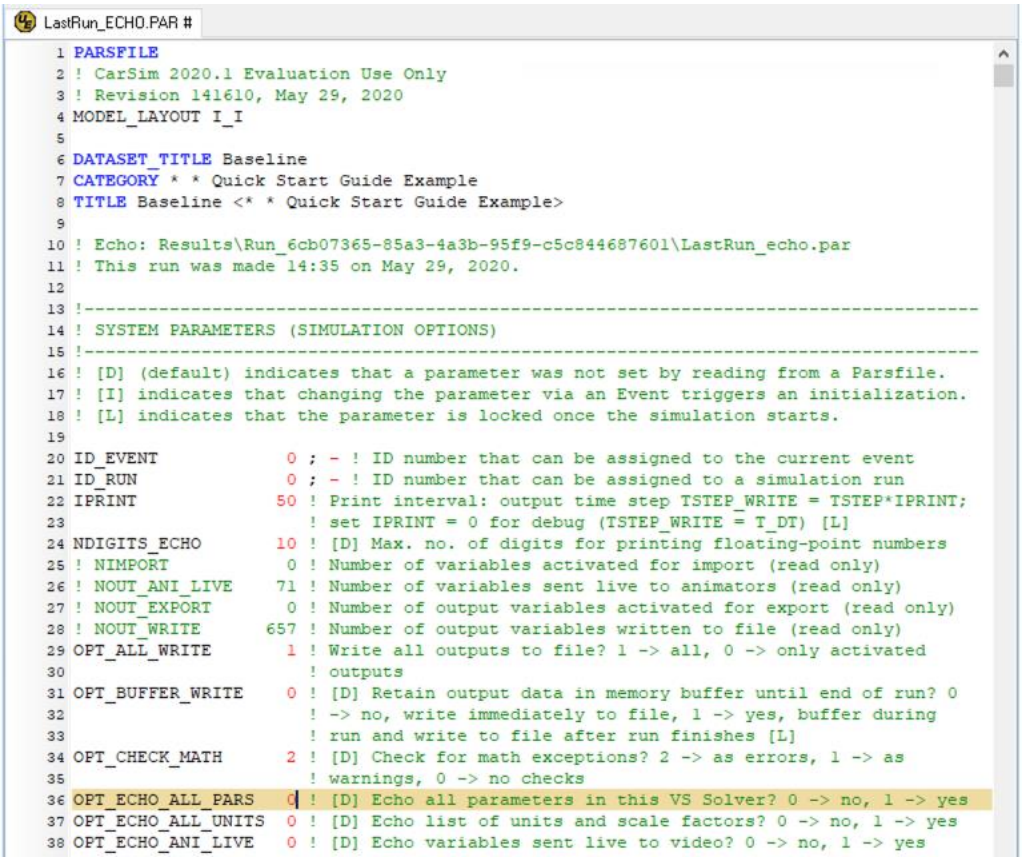

Figure 1. The top part of an example Echo file.

Overall, the Echo file is always the most current reference document available that lists and defines all keywords recognized by the VS Math Model.

To change any of these parameters you may enter the keyword and new value in a miscellaneous data field in the GUI. For example, Figure 2 shows how the parameter `OPT_ECHO_ALL_PARS` is set to 1 using the miscellaneous data field that can be shown in the bottom of the **Run Control** screen.

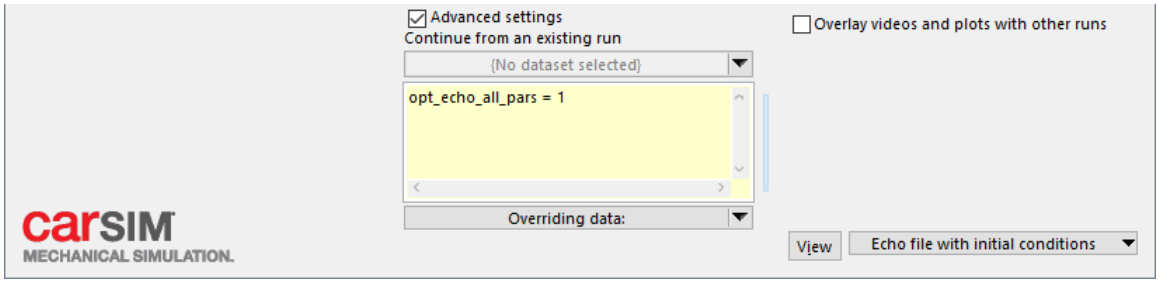

Figure 2. Set a parameter in a misc. yellow field with a keyword and a value.

Nearly all parameters in a VS Math Model whose keywords start with `N` or `OPT_` are integers. The Echo file gives a short description that explains the significance of the integer values (typically 0 indicates an option is disabled and 1 indicates the option is enabled). By convention, the first value described is usually the default value that will be used if no information is provided in the inputs

to the VS Math Model. For example, the option OPT_ALL_WRITE (lines 29 and 30, Figure 1) defines the value of 1 first, indicating that this is the default value that is used if the inputs to the VS Math Model do not provide a value for this parameter.

Titles and Messages

Each run has a title that is written in the Echo file and VS or ERD file header file. Titles and messages are also written into the log file. Titles and messages are simple strings of alphanumeric characters that are not used directly by the VS Math Models, but which serve to document the run conditions.

Lines in Parsfiles that specify strings have a different interpretation than other lines. In most cases, the first word is the keyword that identifies the title, and the remainder of the line is the title. Any spaces between the keyword and the first character of the title are removed, and any spaces the end of the line are removed.

It is also possible to specify titles and messages using a mixture of strings identified by double quote marks (") combined with printed values of symbolic expressions, as described in the **Help** document **Reference Manuals > VS Commands**.

CATEGORY

Definition:

> CATEGORY is a title written into the header of the ERD or VS file. It appears near the top of the Echo file (line 7, Figure 1).

Valid values:

> VS Math Models do not use this string other than to show it in the Echo file for documentation purposes. It has a length limit of 127 characters.

Where to set:

> This is normally a copy of the category of the **Run Control** dataset used to make the run; it is automatically written with the CATEGORY keyword in the Parsfile for the **Run Control** dataset. If the run is created from the VS Browser, it is very difficult to override the text from the **Run Control** dataset.

DATASET_TITLE

Definition:

> DATASET_TITLE is a title written into the header of the ERD or VS file and used in the automatic labeling of plots overlaid from different files. It appears near the top of the Echo file (line 6, Figure 1).

Valid values:

> VS Math Models do not use this string other than to show it in the Echo file for documentation purposes. It has a length limit of 127 characters.

Where to set:

This is normally a copy of the dataset title of the **Run Control** dataset used to make the run; it is automatically written with the `DATASET_TITLE` keyword in the Parsfile for the **Run Control** dataset. If the run is created from the VS Browser, it is very difficult to override the text from the **Run Control** dataset.

TITLE

Definition:

`TITLE` is a title written into the header of the ERD or VS file and in the Echo file.

Valid values:

VS Math Models do not use this string. It has a length limit of 255 characters.

Where to set:

The VS Browser generates this automatically by combining the category name and dataset name for the **Run Control** dataset used to make the run; it is automatically written with the `TITLE` keyword in the Parsfile for the **Run Control** dataset. If the run is created from the VS Browser, it is very difficult to override the text from the **Run Control** dataset.

The VS Browser combines the category string *category* and the dataset string *dataset* as: *dataset* <*category*> (line 8 in Figure 1, page 6).

Specifying Information for the Echo Files

Notice that some of the parameters shown in Figure 1 have names that include `_ECHO`. These parameters help determine the information that will be shown in the Echo file.

NDIGITS_ECHO

Definition:

`NDIGITS_ECHO` is the maximum number of significant digits that will be used to write double-precision numbers into an Echo file.

Valid values:

The default value is 10. For typical vehicle models, the number should not be less than 6 (the minimum allowed is 4). Values can be as high as 15 to obtain maximum precision, but values higher than 12 are not needed for any situations seen to date.

Where to set:

Anywhere before the run starts.

VS Math Models represent all floating-point variables internally using double precision (64 bits). However, numerical values are written in the database and Echo files from the solvers with less precision, to make the files more readable. (For example, compare `"302"` to `"301.9999999999573245"`.) Most variables used to describe vehicle properties are typically measured with a precision involving 6 significant digits or fewer.

Along with the database files, all parameters in the VS Math Model are written in Echo and End files generated for each run. All double-precision numbers are written with a C format string with the form: "%.*n*g" where *n* is the specified number of significant digits.

The default precision is fine for most parameters and tables. However, sometimes 3D road tables are obtained using global coordinates involving thousands of kilometers, with the variations involving tenths of millimeters. These numbers can have as many as 12 significant digits. To avoid round off error in the Echo file, set NDIGITS_ECHO to 12 or higher when using data values that truly require full double precision.

OPT_ECHO_ALL_PARS

Purpose:

> Optionally write all parameter keywords in the Echo file, even if they are not normally seen.

Valid values:

> 0 → show only the parameters and configurable functions that were used for the simulation run (default)
>
> 1 → show all parameters and configurable functions that could possibly be used in a run with the specified VS Math Model, including parameters that are normally hidden

Where to set:

> Specify the keyword and a value in a miscellaneous field.

Normally, the Echo file does not list keywords that were not used in the current run, or support parameters that determine context for reading indexed parameters. For example, all CarSim Math Models support active steering on the rear axle and have built-in parameters to describe the rear-wheel steering. In most cases, the parameter OPT_STEER_EXT is set to zero for the rear axle (not steered), so the steering parameters are not used and are not visible in the Echo file unless OPT_ECHO_ALL_PARS = 1.

There are other variables that are used, but are either read-only, or are used to set the scope for reading indexed parameter. For example, Figure 3 shows part of an Echo file generated with OPT_ECHO_ALL_PARS set to 1. In this example, the section with the title MODEL PARAMETERS would normally not be visible, with parameters that are normally hidden such as IAXLE, IUNIT, etc.

Note that parameters that are normally hidden are all shown as comments (the lines begin with the comment character '!'). This is done to avoid conflicts for advanced users who may use an Echo file as an input for a VS Math Model to make a new simulation.

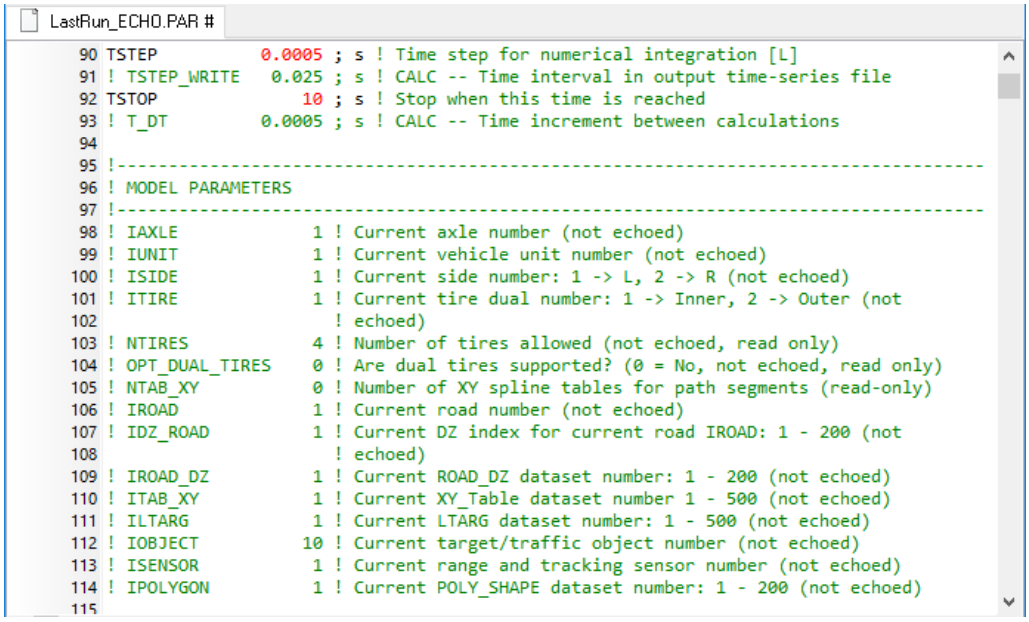

Figure 3. An Echo file when OPT_ECHO_ALL_PARS is enabled.

OPT_ECHO_ALL_UNITS

Purpose:

Optionally write definitions of all units that are recognized by the VS Math Model.

Valid values:

0 → do not show definitions of units (default)

1 → show definitions of all user-display units that were installed for the current run

Where to set:

Specify the keyword and a value in a miscellaneous field.

A VS Math Model maintains two major sets of units:

1. All calculations during a run involve *internal units* that require no conversions. Lengths are meters, angles are radians, accelerations are m/s^2, etc.
2. *User display units* are accepted for inputs and shown for outputs.

Any scaling that is needed to convert between user display units and internal units is done automatically. For example, lengths with units of millimeters are divided by 1000 on input and multiplied by 1000 on output.

Figure 4 shows the start of an Echo file when `OPT_ECHO_ALL_UNITS` = 1. The information in the Echo file gives the names and scale factors for all user display units. For example, line 22 shows that the conversion for variables with user-display units "`deg`" has a scale factor of 57.2957795131; any variables with units of `deg` are converted from internal units (`rad`) by multiplying by 57.2957795131.

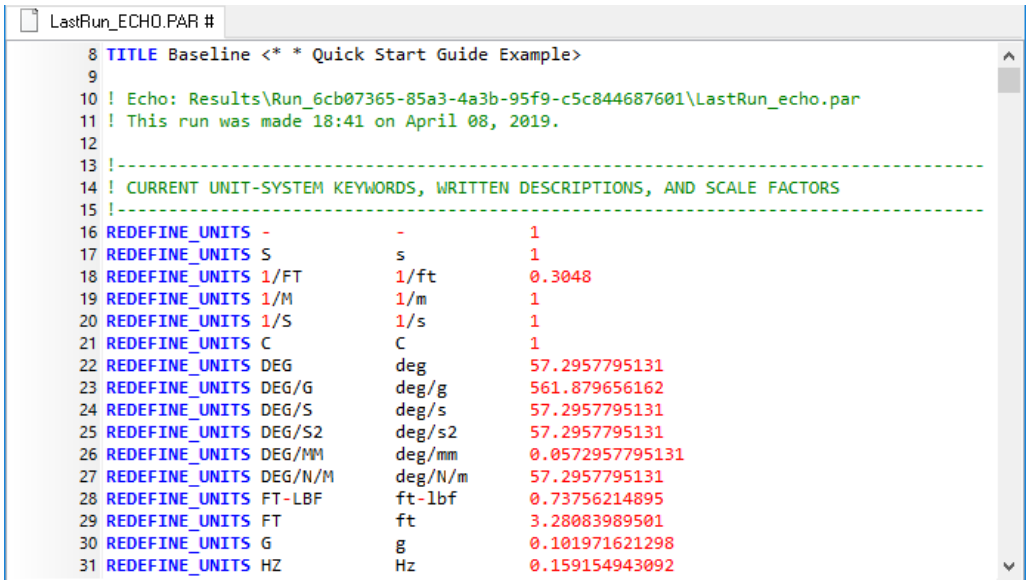

Figure 4. Echo file with display of all installed user units.

Note Each line in this section begins with the keyword `REDEFINE_UNITS`. This is a VS Command, documented in the **Help** document **Reference Manuals > VS Commands**, for specifying user-display units. The `REDEFINE_UNITS` commands are inserted at the very top of the Echo file, because some advanced users might send an Echo file to a VS Math Model to duplicate or continue a run.

OPT_ECHO_ANI_LIVE

Purpose:

Optionally list output variables that were activated for live animation.

Valid values:

0 → do not list output variables (default)

1 → list all activated output variables for live animation

Where to set:

Specify the keyword and a value in a miscellaneous field.

VS Math Models have many built-in variables that can be written to output files for post-processing animation and plotting. When used in real-time environments, the same variables can be sent to one or more animators for live animation as the run proceeds in real time. A later section (page 20) describes how potential output variables are activated for a run.

The keyword `OPT_ECHO_ANI_LIVE` is used to list all of the output variables that were activated for live animation (Figure 5). Each line has the keyword `ANIMATE`, then the output variable name, then the value at the start of the run, and finally by a description of the variable and the index number in the current run. A list of these variables is also obtained automatically when sending

data to the VS Math Model. The browser scans the Parsfiles for animator requirements and generates `ANIMATE` commands for the required variables.

```
5441 !-------------------------------------------------------------------------------------
5442 ! VARIABLES BROADCAST FOR LIVE ANIMATION
5443 !-------------------------------------------------------------------------------------
5444 ANIMATE XO          0 ! #1. X coordinate, vehicle origin (m)
5445 ANIMATE YO          0 ! #2. Y coordinate, vehicle origin (m)
5446 ANIMATE ZO -0.007430279269 ! #3. Z coordinate, vehicle origin (m)
5447 ANIMATE X_L1 0.000784762778 ! #4. X coordinate, wheel center L1 (m)
5448 ANIMATE Y_L1   0.8375 ! #5. Y coordinate, wheel center L1 (m)
5449 ANIMATE Z_L1 0.3175687733 ! #6. Z coordinate, wheel center L1 (m)
5450 ANIMATE YAW_L1 -1.015042913e-15 ! #7. Yaw, wheel L1 (deg)
5451 ANIMATE ROT_L1 0.0003843046051 ! #8. Wheel L1 rotation (rev)
5452 ANIMATE ROLL_L1 1.590277341e-15 ! #9. Roll (Euler), wheel L1 (deg)
5453 ANIMATE X_R1 0.000784762778 ! #10. X coordinate, wheel center R1 (m)
```

Figure 5. List of output variables broadcast for live animation.

OPT_ECHO_DEFAULT

Purpose:

> Disable the identification of default values that were not set by reading an input file with the indicator [D] in the Echo file, as shown earlier in Figure 1 (page 6).

Valid values:

> 1 → use [D] in the commented description of a parameter to indicate a default value that was not set by reading from an input file
>
> 0 → do not use [D] in the Echo file

Where to set:

> Specify the keyword and a value in any miscellaneous field.

VS Math Models include many parameters and properties of Configurable Functions that are not explicitly set in most runs. They typically show [D] in the commented description. However, there are special cases where the presence of the indicator interferes with text comparisons of Echo files.

For example, when a run is continued by reading data from an existing Echo file, all parameters are read from that file and will therefore not have the [D] indicator. A text comparison of that Echo file with another Echo file will show that nearly all lines of text differ. In that usage case, the differences are avoided by setting `OPT_ECHO_DEFAULT` to 0 for the runs where Echo files will be compared.

OPT_ECHO_ICS

Purpose:

> Optionally list all state variables in the VS Math Model, along with their initial conditions (ICs) at the start of the run.

Valid values:

0 → do not list state variables (default)

1 → list all state variables along with their current values

Where to set:

Specify the keyword and a value in a miscellaneous field.

The state of a VS Math Model is defined by a set of state variables. During a simulated test, the VS Math Model uses internal equations to calculate new values for the state variables at each time step, as described the *VS Math Models Reference Manual* and other documents.

> **Note** A notable exception to the above description is the VS Math Model constructed in SuspensionSim; it has no differential equations. For each step in the simulation, a SuspensionSim Math Model calculates the static equilibrium state for the model, independently of the state calculated for the previous step. `OPT_ECHO_ICS` is hidden in SuspensionSim and has no effect.

Advanced users can use this listing to get information about the initial or final state of the VS Math Model, depending on whether the Echo or End file is viewed (Figure 6).

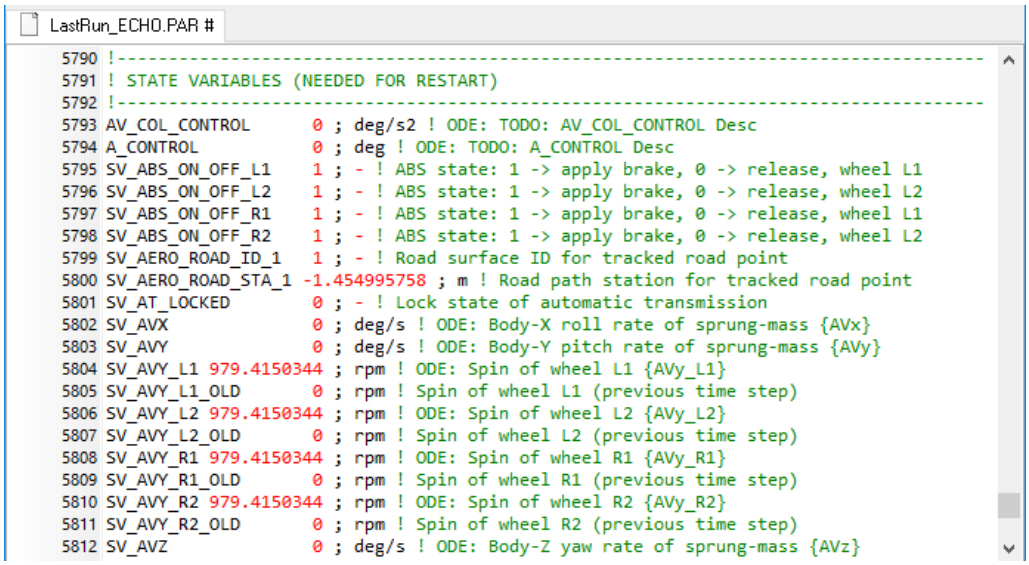

Figure 6. List of state variables and current values.

The End file generated at the end of the run always includes the state variables and their final values, written in alphabetical order. `OPT_ECHO_ICS` is used to provide similar information in the Echo file written at the start of the run.

The keywords for the state variables can be used advanced users to specify initial conditions for the internal VS Math Model, although this is not recommended for normal use.

OPT_ECHO_KEYWORDS_LC

Purpose:

Write the Echo file with most keywords and VS Commands written in lower case (Figure 7).

```
5688 z_profile_constant(2) 0 ; mm ! [D] Constant elevation increment
5689 z_profile_offset(2) 0 ; mm ! Offset added (after gain) to get elevation increment
5690
5691 !-------------------------------------------------------------------------------
5692 ! INITIALIZATION EQUATIONS (APPLIED JUST AFTER INITIALIZATION)
5693 !-------------------------------------------------------------------------------
5694 eq_init road_id_obj(1) = current_road_id;
5695 eq_init road_id_obj(2) = current_road_id;
5696 eq_init road_id_obj(3) = current_road_id;
5697 eq_init road_id_obj(4) = current_road_id;
5698 eq_init road_id_obj(5) = current_road_id;
5699 eq_init road_id_obj(6) = current_road_id;
5700 eq_init road_id_obj(7) = current_road_id;
5701 eq_init road_id_obj(8) = current_road_id;
5702 eq_init road_id_obj(9) = current_road_id;
5703 eq_init road_id_obj(10) = current_road_id;
5704
5705 !-------------------------------------------------------------------------------
5706 ! EQUATIONS OUT (AT THE END OF EVERY TIME STEP)
5707 !-------------------------------------------------------------------------------
5708 eq_out x_obj_1 = x_dm_1;
5709 eq_out y_obj_1 = y_dm_1;
5710 eq_out x_obj_2 = x_dm_2;
```

Figure 7. VS Commands in an Echo file when OPT_ECHO_KEYWORDS_LC = 1.

Valid values:

0 → use ALL CAPS for keywords and VS Commands (default) written in Echo file

1 → write the Echo file with most keywords and VS Commands written in lower case

Where to set:

Specify the keyword and a value in a miscellaneous field.

There are two reasons that this option might be used:

1. Some users prefer lower-case for viewing commands and variable names.
2. If contents are copied from an Echo file and pasted into yellow fields in a VS Browser, lower-case names take up less room, allowing VS commands to be more easily seen.

OPT_ECHO_WRITE

Purpose:

Optionally list output variables that were activated for writing to file.

Valid values:

0 → do not list output variables (default)

1 → list all output variables that were activated for writing to file

Where to set:

Specify the keyword and a value in a miscellaneous field.

VS Math Models have many built-in variables that can be written to output files for post-processing animation and plotting. A later section (page 20) describes how potential output variables are activated for a run.

The keyword `OPT_ECHO_WRITE` may be used to generate a list of the output variables that were activated for writing to file. The list of variables is written near the end of the Echo file written in alphabetical order (Figure 8). Each line has the keyword `WRITE`, then the output variable name, then the value at the start of the run, and finally by a description of the variable and the index number in the current run. For example, line 5798 in the figure shows that the output variable Ay is written to file, that the initial value is close to zero 0, and that it is output variable #6.

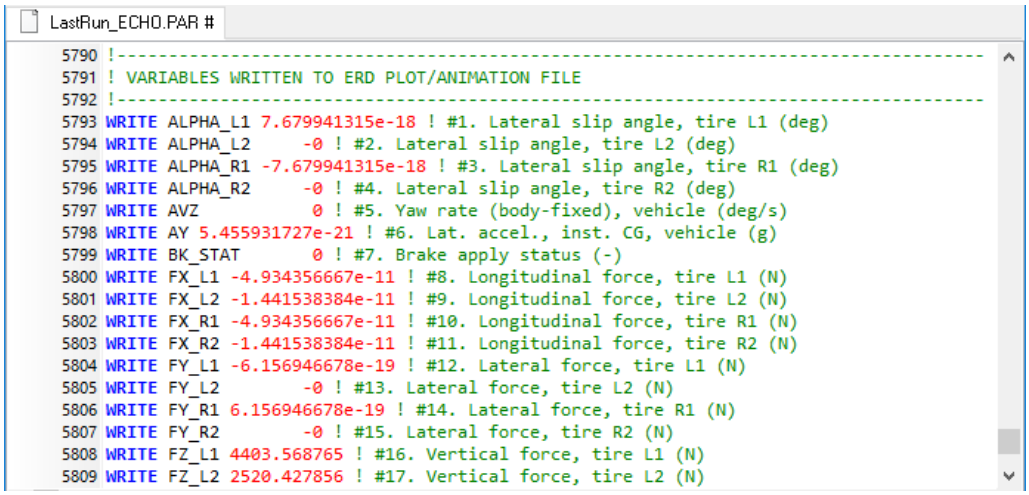

Figure 8. List of output variables written to file for post-processing visualization.

When `OPT_ECHO_WRITE` = 1, the End file (with suffix `_end.par`) has the values of the output variables at the end of the run.

Note This capability of generating a list of all output variables with their instant values at the end of a run is sometimes employed by advanced users to quickly obtain summary results for a run.

Time and Numerical Integration

VS Math Models typically calculate variable of interest from a starting time to a stop time, or until some specified condition is reached and the run is stopped.

Time Range of the Simulation

OPT_STOP

Purpose:

Choose among options for stopping the simulation run.

Valid values:

- < 0 → run until a user-defined stop condition occurs (do not consider TSTOP or SSTOP for stopping the run)
- 0 → stop the run if the simulation time reaches TSTOP (do not consider SSTOP for stopping the run)
- > 0 → In BikeSim, CarSim, and TruckSim, stop the run if the simulation time reaches TSTOP or if the vehicle station reaches SSTOP (a specified distance along a path). Not available in SuspensionSim.

Where to set:

OPT_STOP can be set from drop-list controls on the **Procedures** screen and the **Run Control** screen.

The option where OPT_STOP is set to –1 or another negative value is often called "run forever" and is used for procedures where the run is stopped with a VS Event.

TSTART

Definition:

TSTART is the initial value of the simulation time. The default value is zero.

Where to set:

TSTART can be set from the **Procedures** screen and the **Run Control** screen.

TSTART is normally set to zero. Other values can be used and coordinated with TSTART_WRITE (page 25) to hide initial transient behavior, or to align plots with other simulations or test results, or to continue a run.

In most real-time systems, the start time is zero and this setting is ignored. The TSTART field on the **Run Control** screen is hidden when linked to any RT model dataset.

TSTOP

Definition:

TSTOP is time limit for the simulation

Where to set:

TSTOP can be set from the **Procedures** screen and the **Run Control** screen.

TSTOP is ignored if OPT_STOP is negative.

Time Step and Numerical Integration

VS Math Models with vehicle models compute variables at a fixed time step for use in numerically integrating differential equations. For details, please the *VS Math Models Reference Manual* and the tech memo *Numerical Integration Methods in VS Math Models*.

> **Note** The SuspensionSim VS Math Model is not dynamic; the solver has no differential equations. A time step is used only as a means for showing "time history" plots and scaling the speed for animations viewed with VS Visualizer.
>
> Other than the time step `TSTEP`, the parameters described in this subsection do not exist in the SuspensionSim VS Solver library used to make SuspensionSim VS Math Models.

OPT_INT_METHOD

Purpose:

Select a numerical integration method to solve the internal equations of motion.

Valid values:

–1 → Euler (1st Order Method).

0 → AB-2 (Adams-Bashforth 2nd order method, this is the default).

1 → RK-2 (Runge-Kutta 2nd Order Method).

2 → AM-2 (Adams-Moulton 2nd Order Method).

3 → AM-3 (Adams-Moulton 3d Order Method).

4 → AM-4 (Adams-Moulton 4th Order Method).

Where to set:

`OPT_INT_METHOD` can be set from any of the **Models** screens and the **Preferences** screen.

Two of the above methods involve calculations at the time interval `TSTEP` (Euler and AB-2); the other four (AM-2, AM-3, AM-4, and RK-2) include a second set of calculations at a half step `TSTEP/2`.

Details about these six methods are provided in the tech memo *Numerical Integration Methods in VS Math Models*. The AB-2 method is the preferred method, except for special cases as described in the tech memo.

TSTEP

Definition:

> TSTEP is the major time step used to update all variables in a simulated test. A typical value is 0.0005s for the Euler or AB-2 methods, or 0.001s for an AM or RK integration method.

Where to set:

> TSTEP can be set from the **Run Control** screen, any of the **Models** screens, and the **Preferences** screen.
>
> Be sure that the time step is set to match the type of numerical integration. If you change from AM-2 (which uses a half step) to AB-2 (which uses only full steps), the time step should be cut in half (e.g., change from 0.001s to 0.0005s).

T_DT

Definition:

> T_DT is the interval between calculations of the internal equation of the VS Math Model. It is either set to TSTEP or TSTEP/2, depending on the type of numerical integration set with OPT_INT_METHOD.

Where to set:

> T_DT is calculated; it cannot be set.

This parameter is provided for advanced user to use in equations added with VS Commands that might use finite-difference to calculate time derivatives. It is set to TSTEP if the integration method is Euler or AB-2; for other integration methods it is set to TSTEP/2.

Setting Time Steps, Frequencies, and Numerical Integration

Time steps and frequencies for the VS Math Model can be set from the **Preferences** screen, most **Model** screens, and the **Run Control** screen.

The settings from the **Preferences** screen (Figure 9) provide default values that will be used in all simulations unless they are overridden with settings made elsewhere.

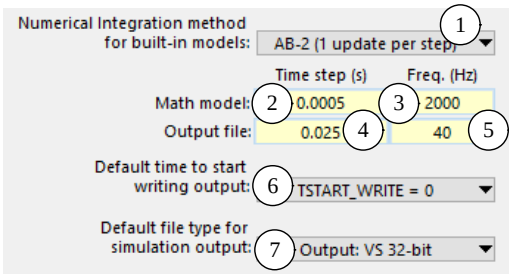

Figure 9. Setting time steps from the Preferences screen.

A default numerical integration option (OPT_INT_METHOD) is selected with the pull-down control (1). The integration method selected here is used for the built-in solvers when there is no link to a **Models** dataset from the **Run Control** screen.

The four numbered yellow fields ((2) – (5)) interact to determine values for the two parameters that control timing: TSTEP and IPRINT. The automatic interactions are provided to support several quick ways for specifying the timing. If you prefer to think in terms of time steps, then you can enter time steps for the VS Math Model ((2) and (4)); the browser will automatically show the corresponding frequencies ((3) and (5)) and enforce an integer relationship IPRINT between the two. If you prefer to enter a value for either frequency, then the browser automatically calculates the corresponding time step, while maintaining an integer relationship between the VS Math Model and output file frequencies.

The **Preferences** screen also has drop-down controls for a standard convention for when to start writing to file (6) (a convention for setting the parameter TSTART_WRITE), and type of output file format (7) (set with the parameter OPT_VS_FILETYPE).

Time steps and frequencies for the VS Math Model and output file can also be set from the **Model** screens (**Models: Self-Contained Solvers**, **Models: Simulink**, etc.), for use with the external model or conditions covered in the Model dataset.

Time steps and frequencies for the VS Math Model and output file can be set from the **Run Control** screen if two checkboxes are checked ((1) and (2), Figure 10). Time steps or frequencies set here will override settings made anywhere else.

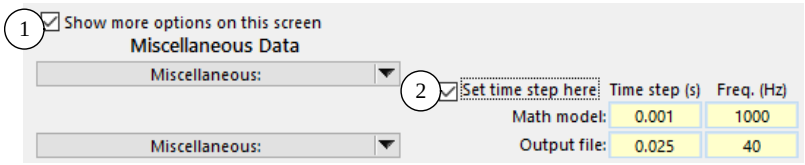

Figure 10. Setting time steps from the Run Control screen.

Writing Outputs to File

The main purpose of a VS Math Model is to calculate time histories of variables of interest. VS Math Models support three formats for these output files: VS, ERD, and CSV. The type of file is specified with the system parameter OPT_VS_FILETYPE. The same file formats are also supported by VS Visualizer.

The VS, ERD, and CSV files can also be automatically processed by the VS Browser to generate other types of files to support third-party post processing software such as MATLAB.

The output variables are also available for export to external software, such as Simulink and LabVIEW. On real-time systems (RT HIL platforms and/or driving simulators), the output variables are also available for streaming in support of live animation.

Selecting Outputs to Write to File

VS Math Models can generate thousands of output variables, and support several options for controlling which of the variables are actually written to file.

When a new run is made, the specifications of which variables should be written can be obtained several ways. When running on Windows with a VS Browser and GUI, the **Run Control** screen has a checkbox **Write all outputs** (1) (Figure 11),

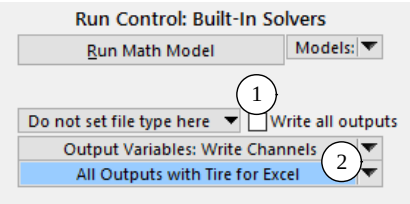

Figure 11. Checkbox and data link on the Run Control screen to specify outputs.

1. If the box (1) is checked, the VS Math Model will write all output variables to the file. This is convenient when investigating behavior for a test in detail; you can view plots of new variables without repeating the run. When the box is checked, the link (2) to a dataset of output variable names is hidden.
2. If the box (1) is not checked, the browser will scan for all references to plots and animations when a run is made and automatically activate all output variables needed to generate the specified animations and plots. The output file will have minimal size for the outputs that will be viewed, saving disk space and minimizing the time needed to load data in VS Visualizer.
3. If the box (1) is not checked, you can link to a dataset (2) that specifies output variables that should always be written into the output file, even if they are not needed for the currently specified animation and plots. The output variable names are typically specified with datasets from the library **I/O Channels: Write**.

Using a Buffer for Real-Time Simulations

VS Math Models support the use of a large memory buffer to save all output variables until the run completes. This option is mainly used in real-time applications where writing to disk during a run can cause a delay that would cause the simulation to drop out of real-time operation.

Another application is simulations where the `SAVE_STATE` and `RESTORE_STATE` VS Commands are used to perform optimizations by moving back in time and trying different parameter options. The buffer can be used so only the final iterations appear in the output files.

The buffer option is enabled using the parameter `OPT_BUFFER_WRITE`. If this option is used, the size of the buffer can be limited with the parameter `NSAMP_BUFFER`.

Parameter Descriptions

IPRINT

Definition:

Integer multiplier to define TSTEP_WRITE interval for writing output variables to file for post-processing.

Valid values:

0 → Write to the output file as often as possible, with time interval TSTEP_WRITE = T_DT

> 0 → Write to the output file at interval TSTEP_WRITE = TSTEP*IPRINT

Where to set:

IPRINT is set automatically whenever TSTEP is set (from the **Run Control** screen, any of the **Models** screens, and the **Preferences** screen).

> **Note** The IPRINT parameter is always set to one in SuspensionSim, meaning TSTEP_WRITE = TSTEP.

NOUT_WRITE

Definition:

Number of output variables that are written to file. This value is calculated and appears near the top of the Echo file (see line 28, Figure 1, page 6).

Where to set:

NOUT_WRITE is calculated; it cannot be set directly.

NSAMP_BUFFER

Definition:

Maximum number of samples of outputs allowed when OPT_BUFFER_WRITE is enabled.

≤ 0 → the VS Math Model will store the entire run based on the stop time TSTOP if OPT_STOP = 1, as described below

> 0 → number of samples stored

Where to set:

Specify the keyword and a value in a miscellaneous field.

This parameter is used only when OPT_BUFFER_WRITE = 1. If OPT_BUFFER_WRITE = 0, NSAMP_BUFFER has no effect and is not visible in the Echo file.

When OPT_BUFFER_WRITE = 1, then all NOUT_WRITE active output variables are saved in memory as the simulation proceeds. If NSAMP_BUFFER is greater than 0, then the total number of values in the buffer is NSAMP_BUFFER*NOUT_WRITE.

If the buffer is limited and the run continues until the buffer is filled, then new samples replace old ones with the effect that the last NSAMP_BUFFER samples are written to file when the run ends. For example, if the buffer holds 100 seconds of data and the run exceeds 100 seconds, then the final 100 seconds are saved and the rest of the calculated outputs are not available for post processing.

The parameter OPT_STOP is used to specify one of two or more modes for stopping the run (page 16). One of these is "run forever" until stopped by a VS Command or other user-defined condition. In this case, the stop time is not known and memory cannot be allocated for the buffer unless NSAMP_BUFFER has a non-zero value. If OPT_STOP is given a negative value (e.g., "run forever" and NSAMP_BUFFER has a value less than 1, the VS Math Model resets NSAMP_BUFFER to 1000 when it allocates memory. This is done after the _echo file is written at the start of the run, so the value of NSAMP_BUFFER shown in the _echo file will not be correct; the correct value is shown in the _end file written when the run terminates.

OPT_ALL_WRITE

Purpose:

Enable/disable the writing of all internally defined output variables to the output file.

Valid values:

1 → write all output variables to file

0 → only write variables that are specified with WRITE and WRT_ commands.

Where to set:

OPT_ALL_WRITE is affected by settings on the **Run Control** screen (see Figure 11, page 20) and the **I/O Channels: Write** screen.

OPT_ALL_WRITE is a VS Extended Assignment Command that sets the value of the parameter with the same name, and clears the list of variables to write in the event that the value of the parameter OPT_ALL_WRITE is changed from non-zero to zero.

Every dataset from the **Run Control** screen applies this command with a value of zero, to ensure that the VS Math Model will initially start with an empty list of output variables. If the checkbox **Write all outputs** (1) is checked (Figure 11, page 20), a line is written at the of the Parsfile that applies this command with a value of 1.

If there is a link to a dataset from the **I/O Channels: Write** library, then the parameter might be set in the linked dataset. There are three options provided by a drop-list control on the **I/O Channels: Write** screen (Figure 12):

1. **Write ALL output variables to file** — the OPT_ALL_WRITE parameter is set to 1 and the VS Math Model will write all output variables.
2. **ADD to the existing list of output variables to write** — no changes are made to the OPT_ALL_WRITE parameter.

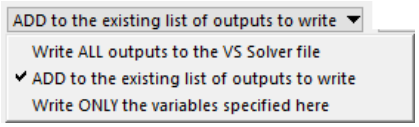

Figure 12. Three options for handling a dataset from the I/O Channels: Write screen.

3. **Write ONLY the variables specified here** — the `OPT_ALL_WRITE` parameter is set to 1, and then set to 0, clearing the list of output variables. The new list contains only the names listed on the **I/O Channels** screen.

> **Note** All outputs are always written in SuspensionSim; the `OPT_ALL_WRITE` parameter does not exist in the SuspensionSim VS Solver library used to make SuspensionSim VS Math Models.

OPT_BUFFER_WRITE

Purpose:

Enable or disable the use of a memory buffer to save all output variables until the run completes.

Valid values:

0 → write output variables to file as the run proceeds (default)

1 → save all output variables to a buffer in memory during the run; write the saved values to file after the run is completed

Where to set:

Specify the keyword and a value in a miscellaneous field. Not necessary if the simulation is done with a dSPACE system that does not have a hard drive.

This option is mainly used in real-time applications where writing to disk during a run can cause a delay that would cause the simulation to drop out of real-time operation. If this option is used, the size of the buffer can be limited with the parameter `NSAMP_BUFFER`.

> **Note** All Parsfiles from the library **Models: Transfer to dSPACE Target** include a line that sets `OPT_BUFFER_WRITE` = 1.

If the buffer is limited, and the run continues until the buffer is filled, then new samples replace old ones with the effect that the last `NSAMP_BUFFER` samples are written to file when the run ends. For example, if the buffer holds 100 seconds of data and the run exceeds 100 seconds, then the final 100 seconds are saved and the rest of the run is not available for post processing.

If (1) the option to use the buffer is specified, and (2) the `OPT_STOP` parameter is set for "run forever" (the stop time is not known), and (3) `NSAMP_BUFFER` has a value less then 1, then the VS Math Model will set `NSAMP_BUFFER` to 1000 when memory is allocated for the buffer (just before the run starts).

OPT_ECHO_WRITE

This enables an option for listing all output variables that will be written by the VS Math Model in the Echo file. Details were provided earlier (see Figure 8, page 15).

OPT_SORT_WRITE

Purpose:

> Choose between two options for setting the sequence of variables written to file or broadcast for live animation.

Valid values:

> 1 → write output variables to file in alphabetical order (default)
>
> 0 → write output variables to file in the order in which they were activated

Where to set:

> Specify the keyword and a value in a miscellaneous field.

The sequence of variables in the output file does not affect the normal behavior of the plotter and animator. However, when creating plots interactively, the plotter shows the variables in the order in which they were read. It is usually more convenient if the variables appear in alphabetical order, regardless of the way they were added when the browser scanned animation and plot datasets.

If `OPT_SORT_WRITE` = 1, the variables are written into the file in alphabetical order.

However, if preparing output files for use with third-party software, it might be necessary to control the sequence in which the variables are written. Set `OPT_SORT_WRITE` to 0 to leave the list in the same sequence in which it was created.

OPT_VS_FILETYPE

Purpose:

> Specify the type of output file.

Valid values:

> 1 → VS format with 64-bit numbers, with a header text file (extension `.vs`) and binary data file (extension `.vsb`) ; this is the default
>
> 2 → VS format with 32-bit numbers, with a header text file (extension `.vs`) and binary data file (extension `.vsb`)
>
> 3 → legacy ERD file format, with a header text file (extension `.erd`) and binary data file with 32-bit numbers (extension `.bin`)
>
> 4 → CSV text file for spreadsheets (extension `.csv`, format is text file with comma-separated values)
>
> 0 → no output file

Where to set:

Drop-down controls for choose the file type are available on the **Preferences** screen and the **Run Control** screen (1) (Figure 13).

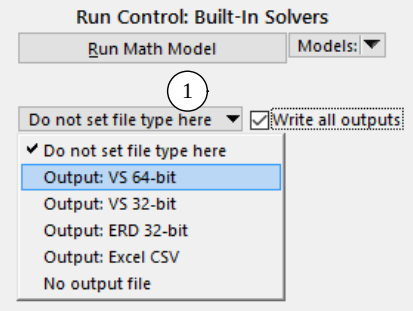

Figure 13. Options on the Run Control screen for the type of output file.

The recommended practice is leave the specification on most **Run Control** datasets as "**Do not set file type here**" such that the setting from the current **Preferences** dataset are used everywhere.

OPT_WRITE

Purpose:

Activate and disable the writing to file as a run proceeds.

Valid values:

1 → write output variables to file whenever T ≥ `TSTART_WRITE` (default)

0 → do not write output variables to file

Where to set:

Specify the keyword and a value in a miscellaneous field. Or, define an equation with a VS Command.

`OPT_WRITE` can be changed dynamically as a run proceeds. It can be changed directly in a Parsfile that is loaded when a VS Event is triggered, or, it can be updated every time step with an equation added with a VS Command. This control can be used in long, complicated scenarios to write to file during portions of interest, and skip the writing during transitions and initializations.

TSTART_WRITE

Definition:

`TSTART_WRITE` is the simulation time at which the VS Math Model starts writing output variables to file.

Where to set:

`TSTART_WRITE` is automatically set to either 0 or `TSTART`, depending on a setting from the **Preferences** screen.

A simulation run starts with the simulated time set to `TSTART`, and then advances in steps. Writing to file starts when T ≥ `TSTART_WRITE`.

If the run starts on a 3D surface that is not flat and level, or if the vehicle is asymmetrically loaded, or if the vehicle is moving with aerodynamic effects, then it will not be in perfect equilibrium at the start of the run and will exhibit transient motions as the suspensions and tires deflect to reach quasi-equilibrium conditions. If these transient motions are not of interest, you can start the run with a negative start time and decline to write to the output file until several seconds have passed.

TSTEP_WRITE

Definition:

> `TSTEP_WRITE` is the time interval used for writing outputs to file. The value is set as described in the above description of `IPRINT`.

Where to set:

> `TSTEP_WRITE` is calculated; it cannot be set.

Connecting with Import and Export Variables

A VS Math Model can be controlled by external software that connects to the model with Import and Export arrays. Import variables are activated for the simulation using the `IMPORT` command, and any output variables can be activated for export using the `EXPORT` command. Please see the *VS Math Models Reference Manual* for more details on how the connections work.

NIMPORT

Definition:

> `NIMPORT` is the number of Import variables that have been activated. This value is calculated and appears near the top of the Echo file (see line 25, Figure 1, page 6).

Where to set:

> `NIMPORT` is calculated by the VS Math Model; it cannot be set directly.

NOUT_EXPORT

Definition:

> `NOUT_EXPORT` is the number of output variables activated for export. This value is calculated and appears near the top of the Echo file (see line 27, Figure 1, page 6).

Where to set:

> `NOUT_EXPORT` is calculated by the VS Math Model; it cannot be set directly.

OPT_IO_SYNC_FM

Purpose:

> Choose between two options for communicating with external software such as Simulink when the Euler or AB-2 integration methods are in use. Not available in SuspensionSim.

Valid values:

1 → communicate with other software with kinematical variables calculated one step ahead of the dynamical variables.

0 → communicate with other software at the start of each time step, with all variables calculated for the same time

Where to set:

OPT_IO_SYNC_FM can be set from any of the **Models** screens.

This option is only available for Euler or AB-2 numerical integration (OPT_INT_METHOD = –1 or 0, both single-step methods), and only when there are both import and output variables (NIMPORT and NOUT_EXPORT are both greater than zero). This allows external software to provide forces and moments as imports to the VS Math Model that were calculated using kinematical variables exported from the VS Math Model, such as with external springs, tires, powertrain, etc.

Please see the *VS Math Models Reference Manual* for more details on how the connections work.

OPT_IO_UPDATE

Purpose:

Choose between two options for communicating with other software, such as Simulink when using an AM or RK numerical integration methods that uses a half-step. Not available in SuspensionSim.

Valid values:

1 → communicate with other software at the interval T_DT if possible (default)

0 → communicate with other software at interval TSTEP

Where to set:

OPT_IO_UPDATE can be set from any of the **Models** screens.

T_DT is a calculated parameter that is either TSTEP or TSTEP/2, depending on the method used for numerical integration. This parameter has no effect if there is no external software communicating with the VS Math Model using Import and Export arrays.

From the point of view of the VS Math Model, both options involve the same numerical integration algorithm and the same time intervals (e.g., full step = 0.001s; half step = 0.0005s). They also produce animation and plot files with the same level of detail. However, from the point of view of external software such as Simulink or LabVIEW, the half-step method updates at twice the frequency.

This option is not recommended for routine use; if variables are to be exchanged rapidly, then the AB-2 integration method is recommended.

Please see the *VS Math Models Reference Manual* for more details on how the connections work.

Error Alerts and the Log File

As a VS Math Model reads input data and runs a simulation, it produces information that can be used to debug problems or confirm that operations went as intended. A log file is normally generated with the suffix `_log.txt` (e.g., `LastRun_log.txt`).

Most errors occur when reading inputs. Therefore, the parameters in this section should be set before reading the bulk of the other Parsfiles. The miscellaneous field on the **Run Control** screen is the last field to be read, so this is not a good place to specify parameters involving error handling or the log file. (If running with external software, the Models dataset is read first, and a miscellaneous field there is a good place to set an error detection parameter. If the Models link is not used, the vehicle dataset is a good place to set the parameters.)

Default values for these parameters can be set globally in the miscellaneous field in the **Preferences** screen. The contents of this field are always written first in the Parsfile sent to the VS Math Models.

OPT_CHECK_MATH

Purpose:

> Generate errors or warnings when VS Commands equations produce non-normal values, such as `infinity` or `NaN` (not a number). A typical scenario would be an equation that produces an unexpected division-by-zero under certain conditions. Subnormal values are permitted.

Valid values:

> 0 → No math checks
>
> 1 → Generate a warning in the log file
>
> 2 → Generate an error and stop the run (default)

Where to set:

> Set in the **Preferences** miscellaneous field or any other miscellaneous field.

OPT_ERROR_DIALOG

Purpose:

> Allow an interactive dialog box to appear with a message if the run is stopped due to an error. Avoid if using automated operations that involve multiple runs.

Valid values:

> 1 → yes, show a dialog box if the run is stopped due to an error (default)
>
> 0 → no, just quit

Where to set:

> Set in the **Preferences** miscellaneous field or a field that is read before other Parsfiles are processed.

OPT_ERROR_DUP_DEF

Purpose:

> Specify how to handle multiple instances of a DEFINE_ VS Command for the same variable.

Valid values:

> 0 → write a warning into the log file
>
> 1 → generate an error and stop the run

Where to set:

> Set in the **Preferences** miscellaneous field or a field that is read before any Parsfile with VS Commands for defining new variables.

If a VS Command for defining a new variable (DEFINE_IMPORT, DEFINE_OUTPUT, DEFINE_PARAMETER, or DEFINE_VARIABLE) attempts to define a variable that already exists, this setting determines the response of the VS Math Model. If not zero, the VS Math Model will generate an error and stop the run. If zero, then the VS Math Model will write a warning into the log file.

If the repeated command is DEFINE_PARAMETER or DEFINE_VARIABLE, and a value is specified, then the new value will be assigned to the parameter or variable when OPT_ERROR_DUP_DEF = 0.

OPT_ERROR_SHOW_ALL

Purpose:

> Control whether multiple pop-up boxes should be used for multiple errors.

Valid values:

> 0 → only show the first error
>
> 1 → also show subsequent errors

Where to set:

> Set in the **Preferences** miscellaneous field or a field that is read before other Parsfiles are processed.

OPT_LOG_DEBUG

Purpose:

> Force the log file to be opened and saved after every message is written. If the VS Math Model crashes, all error messages that were successfully written will be in the log file.

Valid values:

> 0 → write the log file normally, closing it at the end of the run (default)
>
> 1 → close and reopen the log file every time it is updated

Where to set:

> Set in any yellow miscellaneous field that is read before other Parsfiles are processed. For example, the yellow field on the **Preferences** screen will set the value for all runs made when that **Preferences** dataset is active.

If the VS Math Model crashes, the last few lines that should have been written into the log file might be missing. Set `OPT_LOG_DEBUG` = 1 to force the file to be saved any time it is updated. This ensures that all messages will be available for viewing, even if there is a program crash. However, the repeated file manipulation can delay the program slightly.

OPT_LOG_VERBOSE

Purpose:

> Set the amount of information written into the log file.

Valid values:

> 2 → write error messages, information about VS Events, and Parsfile dataset names as they are loaded (default)
>
> 0 → only write error messages into the log file
>
> 1 → write error messages and information about VS Events being activated
>
> 3 → write everything for the default case (value = 2), plus diagnostic information such as every time that the VS Math Model state is saved.

Where to set:

> Set in the **Preferences** miscellaneous field or a field that is read before other Parsfiles are processed.

Log files with all information are normally a few hundred lines long. However, when doing complicated scenarios, especially with RT HIL systems, it is possible to run tests with repeated events over a long period of time (hours, days, weeks, …), generating log files that can become excessive in length when every event or dataset is documented. Set `OPT_LOG_VERBOSE` to zero to avoid this potential problem with long runs involving repetitious events.

OPT_LOG_WARNINGS

Purpose:

> Generate warning messages for indices that are out of range or VS Commands that were not applied.

Valid values:

> 1 → write warnings along with errors in log file (default)
>
> 0 → do not write warnings into log file

Where to set:

Set in the **Preferences** miscellaneous field or a field that is read before other Parsfiles are processed.

The normal behavior of a VS Math Model when encountering a keyword in a Parsfile is to see if it is recognized. If it is not recognized, the line is ignored. If the keyword is recognized, then usually information from the rest of the line is used to update the status of the model in some way. However, there are cases where the rest of the line is not valid, causing the VS Math Model to ignore the input. Here are two examples:

1. Some tables and parameters are tied to indices such as `IAXLE` (axle ID number). If a table or indexed parameter is specified when `IAXLE` is 3 and the vehicle model only has two axles, then the data would be ignored and a warning would be generated.
2. A VS Command to define a new variable would be ignored if the variable has already been defined, and a warning would be generated.

Miscellaneous System Parameters

This section covers the remaining system parameters that appear at the top of an Echo file.

ID_EVENT

Definition:

`ID_EVENT` is a parameter whose value is copied to an output variable every time step. The output variable `ID_Event` can be monitored by advanced users to help track events that were triggered during the simulation.

Valid values:

This number is not used in VS Math Models, so any value can be set.

Where to set:

This is typically set in a VS Event dataset.

ID_RUN

Definition:

`ID_RUN` is a parameter whose value is copied to an output variable every time step. The output variable `ID_Run` can be monitored by advanced users.

Valid values:

This number is not used in VS Math Models, so any value can be set.

Where to set:

This is a legacy parameter that can be set in any miscellaneous field. It would typically be set in a Run Control dataset.

NOUT_ANI_LIVE

Definition:

Number of output variables that are available for animation. This value is calculated and appears near the top of the Echo file (see line 25, Figure 1, page 6).

Where to set:

`NOUT_ANI_LIVE` is calculated to summarize other inputs; it cannot be set directly.

OPT_SKIP_INIT_DYN

Purpose:

Enable or disable the automatic initialization of dynamic variable before writing the Echo file. This parameter is not shown for VS Math Models that do not have dynamics (e.g., SuspensionSim).

Valid values:

0 → Estimate model variables by applying all built-in equations before writing the Echo file. The calculations are repeated using all user-defined VS Commands and Import variable at the first time step, so the results of this initialization mainly affect the display of some calculated values in the Echo file and possibly better initialization of compliance effects. This is the default.

1 → Skip calculations of the built-in equations before writing the Echo file.

Where to set:

`OPT_SKIP_INIT_DYN` can be set from any miscellaneous yellow field.

OPT_SKIP_TSTART

Purpose:

Skip all calculations at the first time (when T = `TSTART`) when running under external software, in case import variables are not ready. Not available in SuspensionSim.

Prior to 2019.0, S-Functions for Simulink skipped all calculations when T = `TSTART` because Import variables were not available yet. With 2019.0, the VS Math Model initialization was improved such that Imports were available, so calculations were enabled. However, some users had placed Simulink "memory block" to avoid Simulink error messages about an algebraic loop. Some of those models no longer worked with the new behavior.

This option to skip the initial calculations was restored for 2019.1, and is enabled with the `OPT_SKIP_TSTART` parameter.

This has no effect unless the VS Math Model is run from external software that uses arrays for import and export variables.

Valid values:

0 → Start normally. This is the default.

1 → Skip calculations until T > TSTART.

Where to set:

OPT_SKIP_TSTART can be set from any miscellaneous yellow field.

OPT_ENABLE_PYTHON

Purpose:

Enable access to embedded Python.

Valid values:

0 → embedded Python is not available (default)

1 → allow Python to be loaded and used to create and apply embedded Python code

OPT_LINEARIZATION

Purpose:

Choose method for defining state variables for the VS Command LINEARIZE. Not available in SuspensionSim.

Valid values:

0 → use all state variables defined with differential equations (default)

1 → activate state variables of interest for use in the linearization

The VS Command LINEARIZE generates linear state equations with matrices A, B, C, and D and writes the matrices to a MATLAB M-File. Perturbing the state of the VS Math Model generates the coefficients in the A, B, and C matrices. If OPT_LINEARIZATION = 0, then the entire Math Model is included. If a non-zero value is set, then only state variables that were activated with the LINEAR_SV command are used to perturb the system.

OPT_PAUSE

Purpose:

Set option to leave the VS Math Model open after a run completes when using a built-in Windows-based solver.

Valid values:

0 → exit the VS Math Model at the completion of a run (default)

1 → leave the progress indicator visible when making a run while using a Windows built-in solver

When running a VS Math Model from the Windows browser, the browser shows a progress indicator with the name of the run and the input and output files. In normal operation, the indicator disappears as soon as the run completes. When OPT_PAUSE = 1, the visual display remains until you click the **Stop** button. This feature is used to help make screen copies for documentation and for debugging.

RTIME

Definition:

> RTIME is the ratio of computation time to simulated time. A value less than 1.0 indicates the VS Math Model ran faster than real time.

Where to set:

> RTIME is calculated by the VS Math Model to summarize computational performance; it cannot be set directly.

This calculated metric is written in the _end file and the log file. The elapsed time obtained from the Windows OS is not very accurate for short runs, but for longer runs, this ratio gives a good estimate of the program efficiency. For example, a ratio of 0.2 means the program took 0.2 seconds for each simulated second, running five times faster than real time.

The RTIME parameter is intended for use with VS Math Models running under Windows OS. On real-time systems, it is hidden because the Windows-based clock is not valid. On these systems, runtime efficiency is usually obtained with tools outside the VS Math Model that are part of the real-time computer system.

TSTART_T_EVENT

Definition:

> TSTART_T_EVENT is an offset subtracted from T to define an event-based clock with current relative time T_EVENT.

Where to set:

> TSTART_T_EVENT is handled like any other parameter and can be assigned a new value at any time. Normally, this would be in a Parsfile that is loaded when a VS Event is triggered. A checkbox on the Events screen named **Reset all control clocks** does this automatically.

< This page is intentionally blank >

Mechanical Simulation

755 Phoenix Drive, Ann Arbor MI, 48108, USA

Phone: 734 668-2930 • Fax: 734 668-2877 • Email: info@carsim.com

carsim.com