Setting Up Import and Output Variables

VS Math Models can be launched and controlled by the VS Browser such that no other software is needed. The main product of the simulation activity is an output file with time histories of variables of interest that can be plotted or used to generate animations with VS Visualizer. Outputs can also be written automatically into files that can be opened in Excel (and other spreadsheet programs) or MATLAB.

VS Math Models for vehicles can also be launched and controlled from other simulation environments such as MATLAB, Simulink, LabVIEW, ASCET, FMI-compatible software, and custom software. Most of the external software tools transfer information to and from a VS Math Model using arrays of Import and Export variables. The VS Math Models support hundreds of variables that can be activated for Import, and thousands that can be activated for Export.

Note The SuspensionSim product includes the **I/O Channels Write** library and supports the export of data to Excel and MATLAB. However, it does not include the libraries used to connect with Simulink and other time-domain simulation environments that may be used with BikeSim, CarSim, and TruckSim.

Output Files

Each VS Math Model includes a number of output variables that can be written to file for later plotting and animation. For example, a four-wheeled CarSim model has a minimum of about 600 built-in output variables. If the model is extended with motion sensors, moving objects, ranging and detection sensors, etc., the number of available output variables can increase to tens of thousands. Vehicle models with trailers start with many more built-in variables. Users may also define new output variables with VS Commands, further increasing the number.

A VS Math Model can be set to write time histories for all existing output variables to file. This option can be set for any simulation using a checkbox on the **Run Control** screen ② (Figure 1). If the box is not checked, then output variables are written only if specified for plots or animations, or if added explicitly with a linked dataset ③ from the **I/O Channels: Write** library.

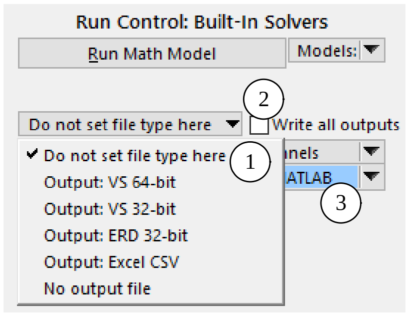


Figure 1. Checkbox "Write all outputs" on the Run Control screen.

VS Math Models support four types of output files ①:

1. VS 64-bit binary
2. VS 32-bit binary
3. ERD 32-bit binary (legacy)
4. CSV (comma-separated variable) text, compatible with spreadsheet programs

All four of these formats are fully supported by VS Visualizer. The main visible differences are that 64-bit binary files are twice as big as 32-bit binaries, while 32-bit files do not keep enough resolution for GPS latitude and longitude and possibly a few other output variables. CSV files are the largest and are slower to load; however, this is often not noticeable on modern computers. The main limit of the CSV file is that is has minimal label information, with no information about units. The lack of information beyond the name of the output variables prevents CSV files from being used in some advanced applications.

The type of file can be set on the **Run Control** screen with a drop-down list control ① (Figure 1). More commonly, it is set with similar controls from the most recently viewed **Preferences** dataset, which is used as the default for all simulations in the current database.

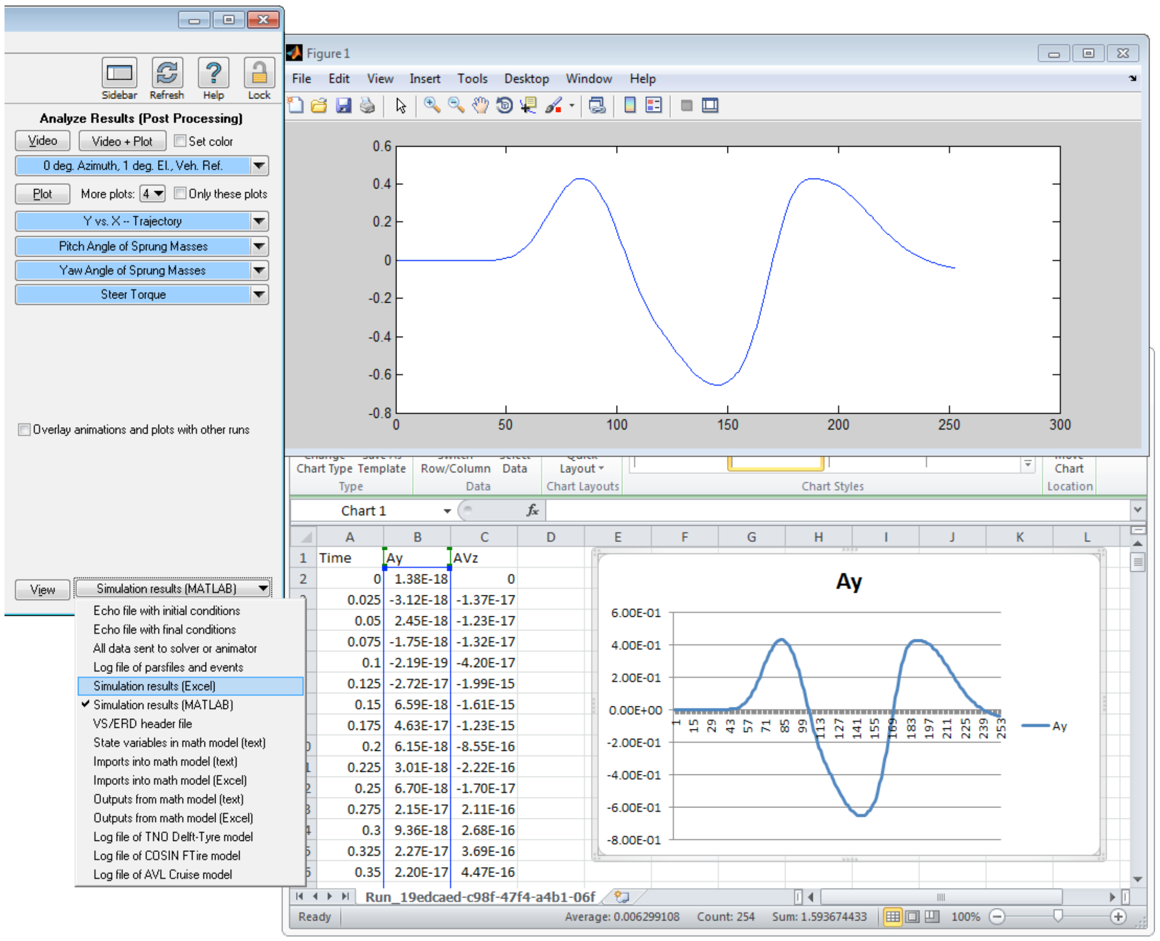
VehicleSim products have options for automatically copying output data from a native output file to files that can be used in MATLAB. When you click a button to run the simulation from the **Run Control** screen, or pass control to Simulink, the VS Browser automatically sets up files needed to run the simulation. It also runs the simulation, either directly (if no external software is used), or by connecting with external software such as Simulink. The VS Math Model sends a signal to the VS Browser when the run terminates.

When the VS Browser receives the signal that the simulation has terminated, it scans the input to see whether MATLAB files were requested. If so, the VS Browser launches a utility program called the VS/ERD File Utility to read the contents of the native output file and write a MATLAB binary file (file type = MAT). Finally, the VS Browser refreshes the screen to enable plotting, video, and viewing of machine-generated files, including output files for MATLAB. The creation of the MAT file occurs so rapidly it is normally not noticed. (It takes about a tenth of a second.)

The option to make a MATLAB file is set up by linking to a dataset from the **I/O Channels: Write** library, as described in the next subsection.

Although a VS Math Model can generate a CSV file directly, the VS/ERD File Utility can also copy information from a VS or ERD file to generate a CSV file with a subset of the contents, as is done for the MATLAB file

CSV and MAT files can be viewed from the **Run Control** screen using the **View** button in the lower right corner (Figure 2).



Sidebar
Refresh
Help
Lock
Analyze Results (Post Processing)
Video
Video + Plot
Set color
0 deg. Azimuth, 1 deg. El., Veh. Ref.
Plot
More plots: 4
Only these plots
Y vs. X -- Trajectory
Pitch Angle of Sprung Masses
Yaw Angle of Sprung Masses
Steer Torque
Overlay animations and plots with other runs
View
Simulation results (MATLAB)
Echo file with initial conditions
Echo file with final conditions
All data sent to solver or animator
Log file of parsfiles and events
Simulation results (Excel)
✓ Simulation results (MATLAB)
VS/ERD header file
State variables in math model (text)
Imports into math model (text)
Imports into math model (Excel)
Outputs from math model (text)
Outputs from math model (Excel)
Log file of TNO Delft-Tyre model
Log file of COSIN FTire model
Log file of AVL Cruise model
Figure 1
File Edit View Insert Tools Desktop Window Help
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
50
100
150
200
250
300
Chart Type Template
Type
Row/Column Data
Data
Layout
Chart Layouts
Chart Styles
Chart
Location
Chart 1
Ay
6.00E-01
4.00E-01
2.00E-01
0.00E+00
-2.00E-01
-4.00E-01
-6.00E-01
-8.00E-01
Ay
Run_19edcaed-c98f-47f4-a4b1-06f
Ready
Average: 0.006299108
Count: 254
Sum: 1.593674433
100%

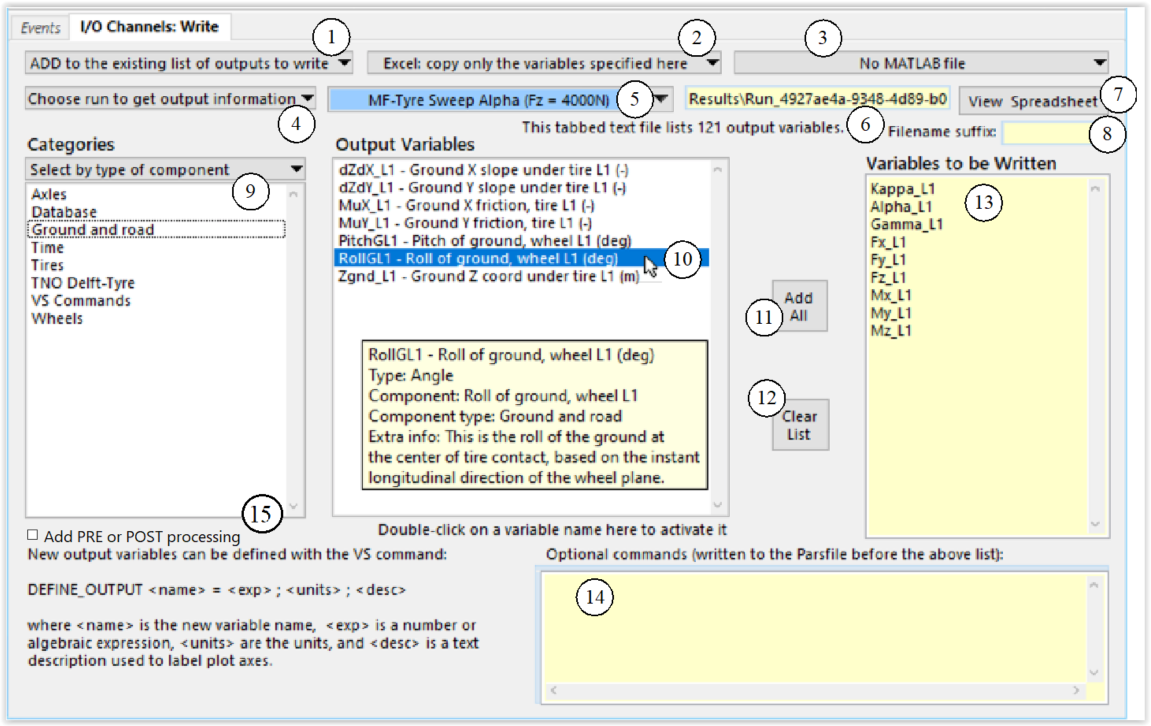
Time	Ay	AVz
0	1.38E-18	0
0.025	-3.12E-18	-1.37E-17
0.05	2.45E-18	-1.23E-17
0.075	-1.75E-18	-1.32E-17
0.1	-2.19E-19	-4.20E-17
0.125	-2.72E-17	-1.99E-15
0.15	6.59E-18	-1.61E-15
0.175	4.63E-17	-1.23E-15
0.2	6.15E-18	-8.55E-16
0.225	3.01E-18	-2.22E-16
0.25	6.70E-18	-1.70E-17
0.275	2.15E-17	2.11E-16
0.3	9.36E-18	2.68E-16
0.325	2.27E-17	3.69E-16
0.35	2.20E-17	4.47E-16

Figure 2. Simulation results from a VehicleSim product can be viewed in Excel and MATLAB.

Note The option to generate a CSV and/or MAT file with a subset of the output variables written to the output file of the VS Math Model only works if the output file type is ERD or VS. If the output type from the VS Math Model is CSV, then no additional files can be made.

I/O Channels: Write

The **I/O Channels: Write** screen (Figure 3) is mainly used to specify variables that should be written to the output file because they will be used in post-processing with third-party software such as Excel or MATLAB. Typically, the dataset will also cause a CSV and/or MATLAB (MAT) file to be created by copying data from the native output file.



ADD to the existing list of outputs to write

Excel: copy only the variables specified here

No MATLAB file

Choose run to get output information

MF-Tyre Sweep Alpha (Fz = 4000N)

Results\Run_4927ae4a-9348-4d89-b0

This tabbed text file lists 121 output variables.

Filename suffix:

Select by type of component

Axles
Database
Ground and road
Time
Tires
TNO Delft-Tyre
VS Commands
Wheels

dZdX_L1 - Ground X slope under tire L1 (-)
dZdY_L1 - Ground Y slope under tire L1 (-)
MuX_L1 - Ground X friction, tire L1 (-)
MuY_L1 - Ground Y friction, tire L1 (-)
PitchGL1 - Pitch of ground, wheel L1 (deg)
RollGL1 - Roll of ground, wheel L1 (deg)
Zgnd_L1 - Ground Z coord under tire L1 (m)

RollGL1 - Roll of ground, wheel L1 (deg)
Type: Angle
Component: Roll of ground, wheel L1
Component type: Ground and road
Extra info: This is the roll of the ground at the center of tire contact, based on the instant longitudinal direction of the wheel plane.

Kappa_L1
Alpha_L1
Gamma_L1
Fx_L1
Fy_L1
Fz_L1
Mx_L1
My_L1
Mz_L1

☐ Add PRE or POST processing

Double-click on a variable name here to activate it

New output variables can be defined with the VS command:

DEFINE_OUTPUT <name> = <exp> ; <units> ; <desc>

where <name> is the new variable name, <exp> is a number or algebraic expression, <units> are the units, and <desc> is a text description used to label plot axes.

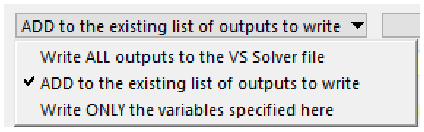
Optional commands (written to the Parsfile before the above list):

Figure 3. I/O Channels: Write screen.

Settings Specific to File Type

The first three controls each apply to a different type of file.

1 Drop-down control with three options for how the variables referenced on this screen interact with other settings involving output variables that might be written to the native output file (Figure 4).



ADD to the existing list of outputs to write
Write ALL outputs to the VS Solver file
✓ ADD to the existing list of outputs to write
Write ONLY the variables specified here

Figure 4. Options for activating output variables for writing.

The second option will add variables specified by name in the scrollable yellow field ⑬. Variables needed for VS Visualizer plotting and video will also be written to file. The third option will clear the list of variables that had been automatically identified for use in plots and video, and create a new list with only the variables named in the yellow field ⑬. Be aware that if the third option (Write ONLY…) is chosen, normal video and plot options will probably not work.

② Drop-down control with three options for making a comma-separated variable (CSV) text file with a subset of the output variables (Figure 5). The CSV format works with Excel and many other programs. If a CSV file is created, it can be opened within Excel using the **View** button on the **Run Control** screen (Figure 2).

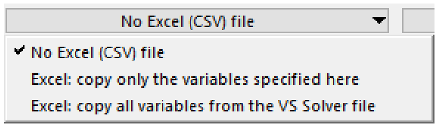

No Excel (CSV) file
✓ No Excel (CSV) file
Excel: copy only the variables specified here
Excel: copy all variables from the VS Solver file

Figure 5. Options for making a CSV file.

The control has three options: no Excel (CSV) file, Excel file with variables specified here (in the scrollable yellow field ⑬), and an Excel file with all variables that exist in the native VS/ERD binary file.

Note VS Math Models can write CSV files as a native output. If the main output is written in CSV format, then no additional files will be made; this control will be ignored.

The option to generate a CSV file using a dataset from this library may be useful for some applications, especially when a CSV file is needed that only has a few of the output variables included in the native file. However, if the main intent is to have all outputs in CSV format, it might be simpler to specify the CSV format from the **Run Control** screen or in the current **Preferences**.

③ Drop-down control with three options for making a MATLAB native binary file (extension MAT) with a subset of the output variables (Figure 6).

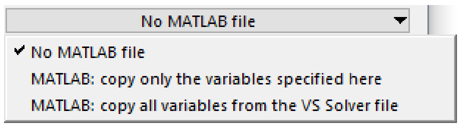

No MATLAB file
✓ No MATLAB file
MATLAB: copy only the variables specified here
MATLAB: copy all variables from the VS Solver file

Figure 6. Options for making a MATLAB binary file.

The three options are the same as described above for Excel files. If a MAT file is created, it can be opened within MATLAB using the **View** button on the **Run Control** screen (Figure 2).

Note If the main output is written in CSV format, then no additional files will be made; this control will be ignored.

Browsing Lists of Available Variables

The three I/O Channel screens (Write, Import, and Export) all support lists of available variables that can be browsed using GUI controls. For all three screens, the information comes from a tabbed text file that is generated dynamically by the VS Math Model. (The same browsing methods are also used for choosing output variables to plot using the **Plot: Setup** library.)

④ Drop-down control to choose between two options for obtaining information about available output variables (Figure 7).

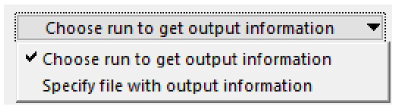

Choose run to get output information
✓ Choose run to get output information
Specify file with output information

Figure 7. Options for getting output information.

The first option arranges the screen to show a potential link to a dataset from the **Run Control** library In this case, use the link to select a run that involves a vehicle model that includes all of the variables you wish to plot.

When you select a dataset, the VS Browser shows all of the output variables that are available in lists on the screen ⑨ and ⑩.

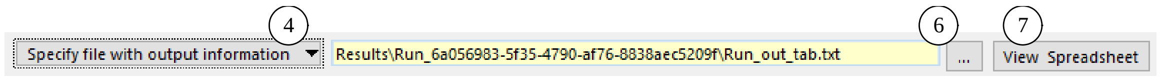
To do this, the VS Browser performs several steps automatically:

1. It creates the VS Math Model that would be used for the simulation, giving it all information from the datasets linked to the selected **Run Control** dataset.
2. It instructs the VS Math Model to create a tabbed text file listing all output variables that are available, given the information available from the **Run Control** dataset. The tabbed text file is written in the `Results` folder associated with the **Run Control** dataset, as indicated in the adjacent yellow field ⑥. The VS Math Model also creates a copy of the same file but with the extension `.xls` (spreadsheet).
3. The VS Math Model is freed from memory.
4. The name of the tabbed text file is placed in the pathname field ⑥.
5. The tabbed text file is scanned to obtain the information shown in the browser lists.
6. The number of available output variables is written in a text message under the pathname field ⑥.

These steps take place very quickly.

When the VS Math Model is used to generate documentation, it does not perform any simulation activities, and can be used even if you do not have license for the specific model of interest. For example, if you have a network license server supporting a Sensor option, you can set up output variables for a vehicle with Sensors without accessing a Sensor license.

The second option arranges the screen to show a larger field for the pathname for a tabbed text file with an adjacent file browser control ⑥ (Figure 8).



Results\Run_6a056983-5f35-4790-af76-8838aec5209f\Run_out_tab.txt

Figure 8. Controls shown for option to specify tabbed text file directly.

⑤ Link to dataset from the **Run Control** library. This link is visible only when the option to choose a run is selected from the drop-down control ④.

When you make the link to a dataset with this control, the VS Browser generates the tabbed text file using the specified **Run Control** dataset as described above. Any time the dataset is visited or the screen is refreshed, the file is read again to provide information about the output variables on the screen.

If you change any of the settings for the selected **Run Control** dataset, or any of the vehicle or procedure settings in datasets linked to the selected **Run Control** dataset, you might want to generate a new tabbed text file. To do so, use the blue link drop-down control ⑤ to re-select the same **Run Control** dataset. Any time you select a **Run Control** dataset with this drop-down control, a new tabbed text file is automatically generated.

⑥ Tabbed text file describing all of the available output variables. This is typically generated by temporarily setting the drop-down control ④ to choose a dataset from the **Run Control** library ⑤ as described above.

The number of output variables is shown immediately underneath this field. In the example, there are 121 output variables (this was a run with the tire tester model).

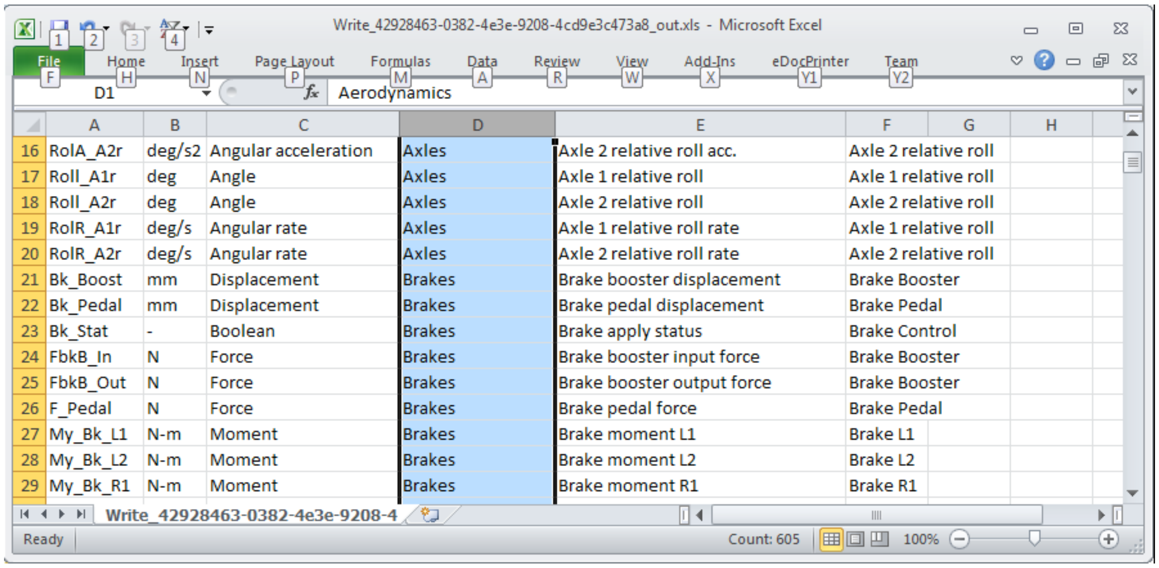
Once the tabbed text file is generated, you can change the drop-down control back to see the full name of the file (Figure 8).

Alert The capability of generating a tabbed text file using the VS Math Model was introduced with CarSim 9.0 in 2014. Older versions of the software made use of static machine-generated tabbed text that do not exist in more recent versions.

If you need to edit an older dataset, you can generate an updated file by linking to a **Run Control** dataset as described above.

⑦ **View Spreadsheet** button. Click to view a spreadsheet with the same information shown on this screen (Figure 9).

The pathname shown in the field ⑥ ends with the text `out_tab.txt`. A second file that ends with the text `out.xls` is also generated and used to support viewing with Excel or other spreadsheet programs using this button.


Write_42928463-0382-4e3e-9208-4cd9e3c473a8_out.xls - Microsoft Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Add-Ins
eDocPrinter
Team
Write_42928463-0382-4e3e-9208-4
Ready
Count: 605
100%

D1 Aerodynamics

	A	B	C	D	E	F	G	H
16	RolA_A2r	deg/s2	Angular acceleration	Axles	Axle 2 relative roll acc.	Axle 2 relative roll		
17	Roll_A1r	deg	Angle	Axles	Axle 1 relative roll	Axle 1 relative roll		
18	Roll_A2r	deg	Angle	Axles	Axle 2 relative roll	Axle 2 relative roll		
19	RolR_A1r	deg/s	Angular rate	Axles	Axle 1 relative roll rate	Axle 1 relative roll		
20	RolR_A2r	deg/s	Angular rate	Axles	Axle 2 relative roll rate	Axle 2 relative roll		
21	Bk_Boost	mm	Displacement	Brakes	Brake booster displacement	Brake Booster		
22	Bk_Pedal	mm	Displacement	Brakes	Brake pedal displacement	Brake Pedal		
23	Bk_Stat	-	Boolean	Brakes	Brake apply status	Brake Control		
24	FbkB_In	N	Force	Brakes	Brake booster input force	Brake Booster		
25	FbkB_Out	N	Force	Brakes	Brake booster output force	Brake Booster		
26	F_Pedal	N	Force	Brakes	Brake pedal force	Brake Pedal		
27	My_Bk_L1	N-m	Moment	Brakes	Brake moment L1	Brake L1		
28	My_Bk_L2	N-m	Moment	Brakes	Brake moment L2	Brake L2		
29	My_Bk_R1	N-m	Moment	Brakes	Brake moment R1	Brake R1		

Figure 9. View from Excel of tabbed text file with available output variables.

Note In addition to the options of using this screen or a spreadsheet program, another option for viewing information about output variables via a simple text file is available via the **View** button in the lower-right corner of the **Run Control** screen.

8 Filename suffix, appended to CSV or MAT files generated using this dataset. When the field is blank, the CSV and MAT file have the same name as the output file generated by the VS Math Model, but with the appropriate extension.

9 **Categories** drop-down control and field. Use the drop-down control to specify how the list of all variables might be sorted (Figure 10). For example, if you choose the option **Select by type of component**, then all of the component types are listed in the field, as shown in Figure 3. If you choose the option **Select by units**, then this list shows all of the units that are used, and the next list shows all of the variables with the currently selected units (km/h, deg, etc.). Click on a category in the list, and those variables associated with the selected category appear in the adjacent list of available variables 10.

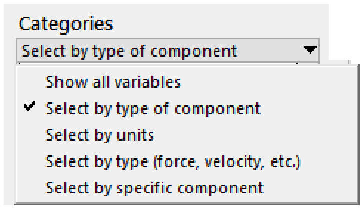

Categories
Select by type of component
Show all variables
✓ Select by type of component
Select by units
Select by type (force, velocity, etc.)
Select by specific component

Figure 10. Options for browsing output variables.

10 **Output Variables** list. This shows the names of all of the output variables that are in the selected category. For example, in Figure 3 the list shows all output variables in the category "Ground and road."

Output variables in a VS Math Model are identified with unique short names that do not contain spaces. For example, the highlighted variable in the list has the short name `RollGL1`.

Along with the short name, each variable has additional labels used for automatically identifying axes and datasets in plots, and supporting interactive browsing. These labels include a long name (e.g., for `RollGL1`, the long name is “Roll of ground, wheel L1”), units (“deg”), a generic name (“Angle”), a component name (also “Roll of ground, wheel L1”), and a component type (“Ground and road”).

If you select an item and then right-click in this list, the VS Browser shows more information about the selected variable (Figure 3).

Notes Some of the right-click descriptions have extra information. For example, the information shown for variable `RollGL1` includes a description of how it is defined mathematically.

When there are groups of similar variables applied for repeated parts, such as the four wheels of this vehicle, only the first variable in the group has the extra information.

You activate a variable from this list by double-clicking on it. When you double-click, the short name is written at the end of the content of the yellow field **Variables to be Written** ⑬.

Specifying Variables for Writing

All of the controls described in the above subsection (④ - ⑩) support interactive browsing of the output variables that are available for a given **Run Control** dataset. However, none of those controls directly control which variables are activated for writing. The scrollable yellow field ⑬ contains a list of output variables of interest. This is the only information used by the VS Browser, VS/ERD Utility, and the VS Math Model to identify variables of interest from a dataset from this library.

⑪ **Add All** button. Click to add the short names for all of the variables shown in the **Output Variables** list ⑩ (based on the selected category ⑨) to the yellow field **Variables to be Written** ⑬.

⑫ **Clear List** button. Click to clear the list of **Variables to be Written** ⑬.

⑬ **Variables to be Written**. This yellow field lists short names of variables that will be included in output files. Variables can be added by double-clicking on names shown in the **Output Variables** list ⑩ or clicking the button **Add All** ⑪. Given that this is an ordinary yellow field, names can be typed in directly, or pasted from the Windows clipboard.

If the same output variable name is listed multiple times, it will only appear once in the output files. If a variable name is specified but the variable is not available, then the name is ignored.

VS Commands

(14) Field for inserting optional VS Commands (or other data). This field is provided as a convenient location for defining new output variables or inserting equations. The VS Commands are all described in the VS Commands manual.

If you define new output variables in this field, they will be included in the tabbed text file if the commands were present when the file was generated. If you change the contents of this field to add more outputs, or change their properties, then you can regenerate the tabbed text file by reselecting a **Run Control** dataset (5) that links to this dataset.

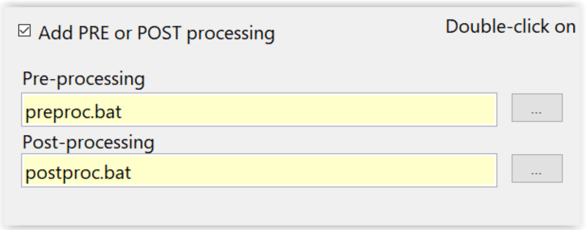
New output variables defined with the VS Command `DEFINE_OUTPUT` will be created, but will not be written to file unless the names are listed in the list of **Variables to be Written** (13), or if all variables are written.

Note An example of defining a new variable and activating it is shown for the **I/O Channels Import** screen (page 19). The same general method is used for output and export variables.

Pre- and Post- Processing Commands

(15) Checkbox for allowing for optional pre-processing and post-processing commands. If this box is checked, the text description for 'DEFINE_OUTPUT' is removed and replaced with fields to hold the pre- and post- processing commands. File browsers are also provided to locate the commands if needed.

The nominal format is *.bat, but any file (with optional parameters) that can be executed from the command line are acceptable. The pre-processing command is executed after the run_all.par has been loaded but before any processing has begun. The post-processing command is executed after the solver has completed operations (but before any I/O Write files have been written). The executable files are located with respect to the "data" directory of the product. (This is the same directory where the simfile can be found.)


☑ Add PRE or POST processing

Double-click on

Pre-processing

preproc.bat

Post-processing

postproc.bat

Figure 11. Options for designating pre- and post- processing operations.

Simulink S-Function Blocks

To work with external software such as Simulink, the VS Math Model is run from a *wrapper* module that connects a VS library used to the other environment. The wrapper communicates with the calling environment in a way that is standard for that environment. Figure 12 shows the relationships between a Simulink model, the VS S-Function wrapper DLL, and a VS library.

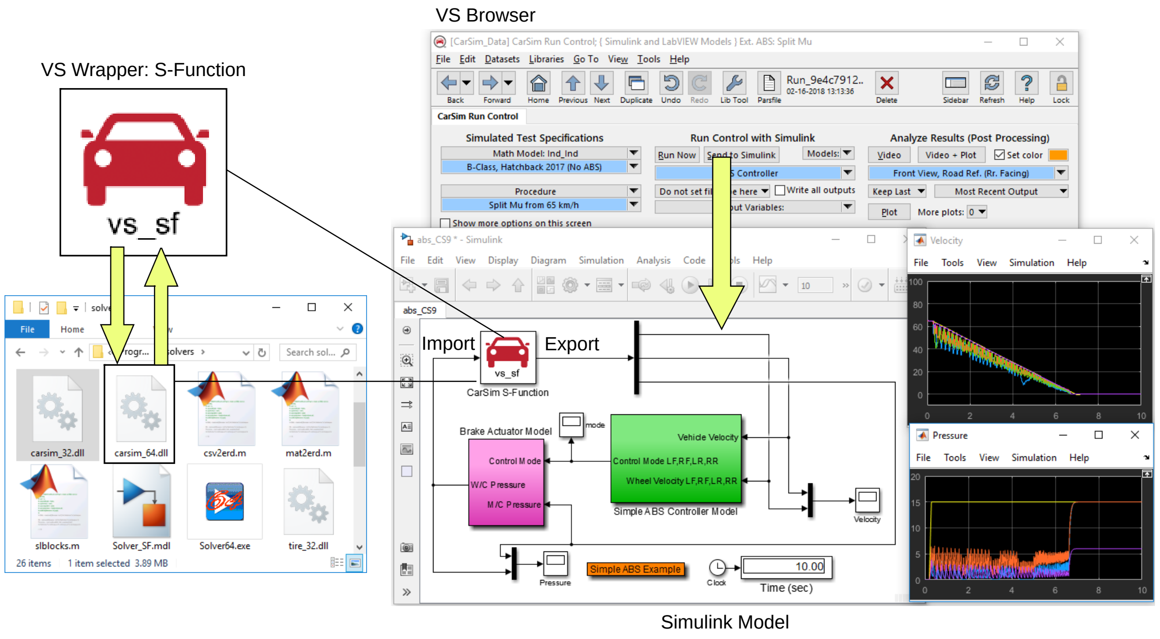


Figure 12. Running a VS Math Model with Simulink.

The VS Browser launches a Simulink model, prepares input files needed to create and run a VS Math Model, and sends commands to Simulink. During the run, there is close communication between Simulink and the wrapper DLL, and between the wrapper and the VS library.

Simulation tools such as Simulink have standard methods for communicating with program modules using arrays of Import variables and Export variables. VS Math Models are compatible with this method: they include hundreds of variables that are referenced in the equations of motions but which are nominally zeroed. They also have thousands of available output variables. Along with the parameters and tables that set properties of the simulated vehicle and test condition, the datasets read by the VS Math Model can include commands that activate variables for Import and Export, as needed to work with external models.

VehicleSim products come with a library of four S-Function blocks. Figure 13 shows the blocks provided in CarSim; similar S-Function blocks are provided with BikeSim and TruckSim.

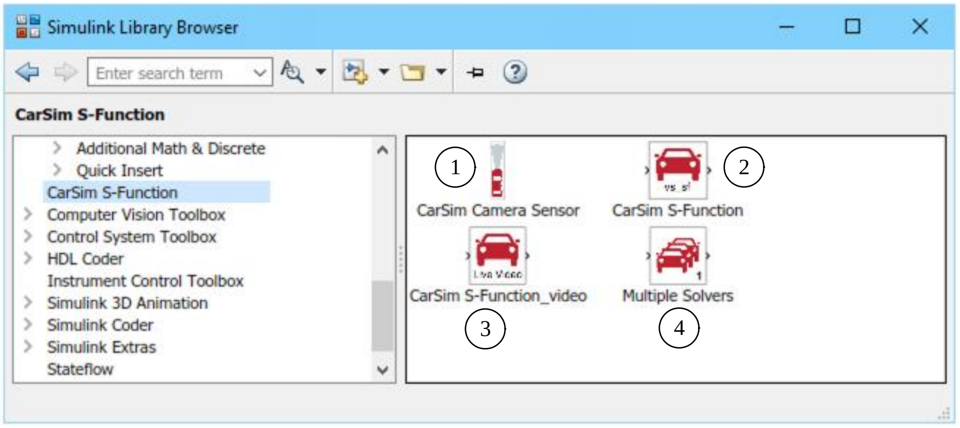


Figure 13. CarSim S-Functions.

Note The S-Function programs are identical in BikeSim, CarSim, and TruckSim; only the graphic image shown within Simulink is specific to the product.

① The Camera Sensor block is a special-purpose S-Function that supports live animation and access to camera sensor outputs from VS Visualizer. For details, please see the Tech Memo *VS Camera Sensor Simulink Block*.

② The S-Function block is the standard VS S-Function used in nearly all example simulations. Import and export ports can be configured as needed, with new ports having any number of specified signals.

③ The S-Function2v block is a variant of the standard VS S-Function that supports live video. Because it establishes unique connections to VS Visualizer, only one instance of this block can be used in a Simulink model

④ The Multiple Solvers S-Function block may be used multiple times in a Simulink model in order to simulation multiple vehicles that interact. Simulink models that use this S-Function should be accessed from the **Tools** > **Parallel Solvers** library screen.

In order for the VS Math Model to communicate properly with the Simulink model, the variables that are imported and exported must match the expectation of the Simulink model.

The easiest way to specify variables for import and export is to use the screens from the libraries **I/O Channels: Import** and **I/O Channels: Export**. If multiple ports are used in Simulink, then they are managed with the library **I/O Channels: Ports**. The links to these datasets are made for Simulink models using the **Models: Simulink** library, which also specifies the Simulink model files and some communication options, as described document *External Models and RT Systems* available from the Help submenu **Help** > **Model Extensions and RT**.

Other Applications for Import and Export Arrays

Import and Export arrays are used for external simulation tools other than Simulink. The same setups are used for LabVIEW, ASCET, FMI/FMU, and custom programs that might be written in MATLAB, Python, and other languages.

In all of these cases, the **I/O Channels: Import** and **I/O Channels: Export** screens are used to configure the VS Math Model to meet the communication requirements of the external simulation tool.

I/O Channels: Export

The output variables that can be written to file for later plotting and animation can also be exported to other simulation environments such as Simulink, LabVIEW, ASCET, or custom software.

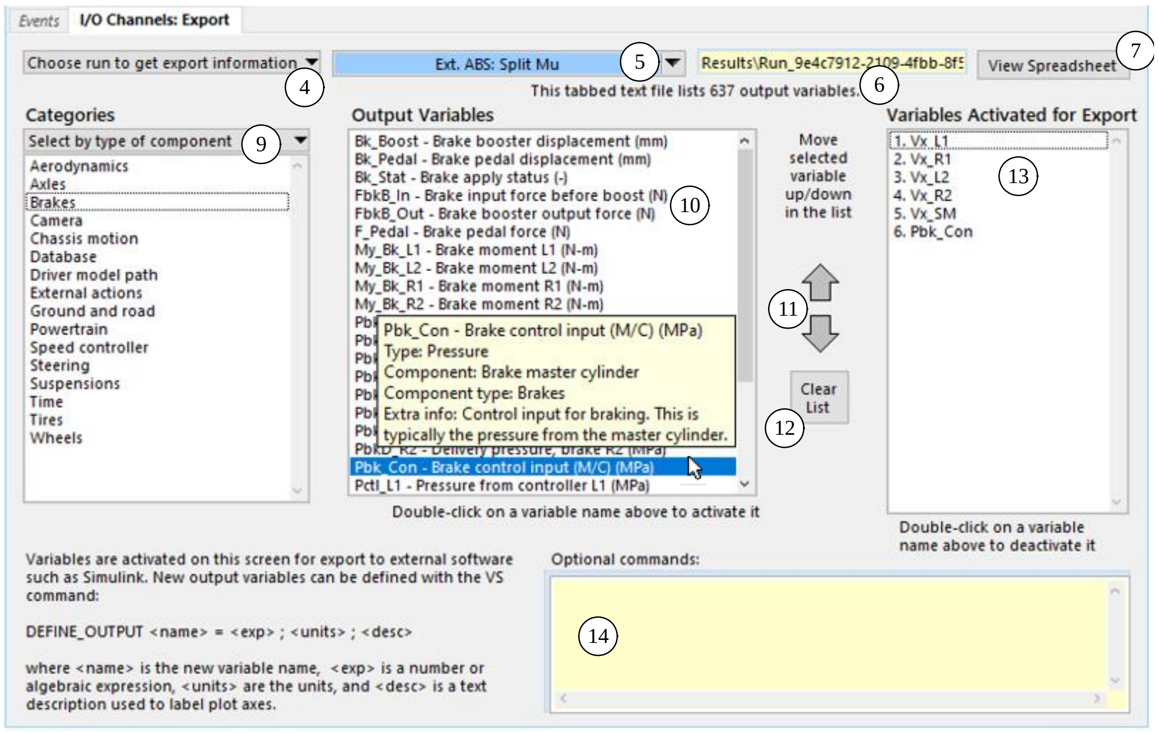
Figure 14 shows the **I/O Channels: Export** screen used to select output variables to export to other software during a simulation. The most typical purpose of a dataset from this library is to set up a *port* with one or more channels (also called signals) that will be sent to an external model from Simulink or LabVIEW. The dataset here must specify the same number of variables as expected in the external model, and the variables must be in the same order expected by the external model.

This screen is similar to the **I/O Channels: Write** screen described earlier. This also involves selecting from available output variables. However, instead of activating them for writing to file, datasets in this library are used to activate them for export. The same output variable can be activated for both writing and export.

This screen also makes use of tabbed text files generated by a VS Math Model to support interactive browsing and viewing from a spreadsheet program. The controls described earlier (④ - ⑩) work on this screen the same as described earlier for the **I/O Channels: Write** screen.

Three controls from the **I/O Channels: Write** screen involving the native file, CSV file, and MATLAB file are not used here.

Two of the controls (⑪ and ⑬) from the **I/O Channels: Write** screen have different behavior in this library, as described below. The differences exist because the list **Variables Activated for Export** is not a yellow field; it is a numbered list ⑬ that shows the short names of the selected output variables. The numbering is shown because it must match the requirements of the linked external model.



Choose run to get export information

Ext. ABS: Split Mu

Results\Run_9e4c7912-2109-4fbb-8f5

This tabbed text file lists 637 output variables.

Categories

Select by type of component

Aerodynamics
Axles
Brakes
Camera
Chassis motion
Database
Driver model path
External actions
Ground and road
Powertrain
Speed controller
Steering
Suspensions
Time
Tires
Wheels

Output Variables

Bk_Boost - Brake booster displacement (mm)
Bk_Pedal - Brake pedal displacement (mm)
Bk_Stat - Brake apply status (-)
FbkB_In - Brake input force before boost (N)
FbkB_Out - Brake booster output force (N)
F_Pedal - Brake pedal force (N)
My_Bk_L1 - Brake moment L1 (N-m)
My_Bk_L2 - Brake moment L2 (N-m)
My_Bk_R1 - Brake moment R1 (N-m)
My_Bk_R2 - Brake moment R2 (N-m)
PbkD_R2 - Delivery pressure, brake R2 (MPa)
Pbk_Con - Brake control input (M/C) (MPa)
Pctl_L1 - Pressure from controller L1 (MPa)

Pbk_Con - Brake control input (M/C) (MPa)
Type: Pressure
Component: Brake master cylinder
Component type: Brakes
Extra info: Control input for braking. This is typically the pressure from the master cylinder.

Double-click on a variable name above to activate it

Variables Activated for Export

1. Vx_L1
2. Vx_R1
3. Vx_L2
4. Vx_R2
5. Vx_SM
6. Pbk_Con

Double-click on a variable name above to deactivate it

Variables are activated on this screen for export to external software such as Simulink. New output variables can be defined with the VS command:

DEFINE_OUTPUT <name> = <exp> ; <units> ; <desc>

where <name> is the new variable name, <exp> is a number or algebraic expression, <units> are the units, and <desc> is a text description used to label plot axes.

Optional commands:

Figure 14. I/O Channels: Export screen.

(11) **Up and down arrow** buttons. You can adjust the order of the Export variables by selecting a variable from the list **Variables Activated for Export** (13) and then using the arrows to move the name up or down in the list. This affects the numbers associated with the variables, as needed to match the requirements of external software such as Simulink where only numerical indices are used to identify the variables received from the VS Math Model.

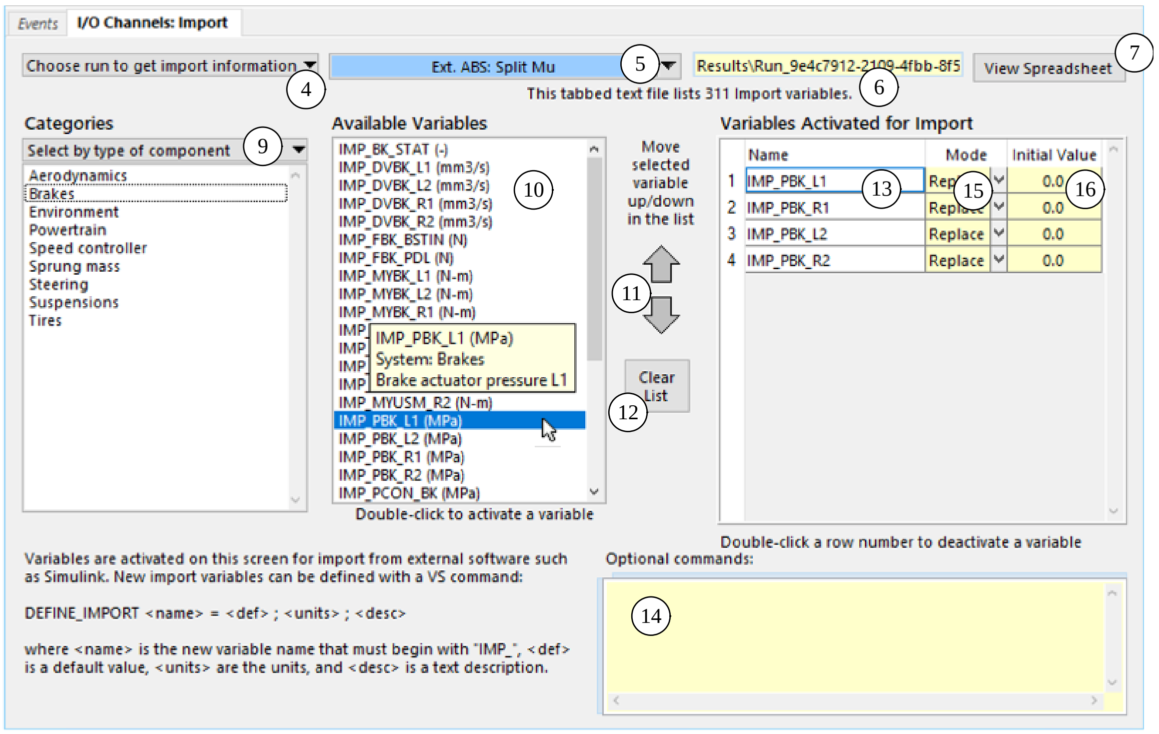
(13) **Variables Activated for Export**. This list shows variables that will be sent to external software as a run proceeds. The external software will identify the variable by the index shown for each name in the list (1, 2, 3, …). The order of the rows (and therefore the indexing) can be adjusted using the up and down arrow buttons (11).

Double-click on a variable to remove it from this list.

An output variable may be added multiple times to this list, as might be needed to work with existing external models. In this case, the same variable value from the VS Math Model is copied into the export array multiple times.

I/O Channels: Import

Use the **I/O Channels: Import** screen (Figure 15) to activate Import variables and specify how they are combined with internal variables in the math model. The typical purpose of a dataset from this library is to set up a *port* with one or more channels (also called signals) that will be provided by an external model from Simulink, LabVIEW, or other external software. Two critical factors in making the connection are:


Events | I/O Channels: Import

Choose run to get import information (4) | Ext. ABS: Split Mu (5) | Results\Run_9e4c7912-2109-4fbb-8f5 (6) | View Spreadsheet (7)

This tabbed text file lists 311 Import variables.

Categories

Select by type of component (9)

Aerodynamics
Brakes
Environment
Powertrain
Speed controller
Sprung mass
Steering
Suspensions
Tires

Available Variables (10)

IMP_BK_STAT (-)
IMP_DVBK_L1 (mm3/s)
IMP_DVBK_L2 (mm3/s)
IMP_DVBK_R1 (mm3/s)
IMP_DVBK_R2 (mm3/s)
IMP_FBK_BSTIN (N)
IMP_FBK_PDL (N)
IMP_MYBK_L1 (N-m)
IMP_MYBK_L2 (N-m)
IMP_MYBK_R1 (N-m)
IMP_MYUSM_R2 (N-m)
IMP_PBK_L1 (MPa)
IMP_PBK_L2 (MPa)
IMP_PBK_R1 (MPa)
IMP_PBK_R2 (MPa)
IMP_PCON_BK (MPa)

IMP_PBK_L1 (MPa)
System: Brakes
Brake actuator pressure L1

Double-click to activate a variable

Move selected variable up/down in the list (11)

Clear List (12)

Variables Activated for Import

	Name (13)	Mode (15)	Initial Value (16)
1	IMP_PBK_L1	Replace	0.0
2	IMP_PBK_R1	Replace	0.0
3	IMP_PBK_L2	Replace	0.0
4	IMP_PBK_R2	Replace	0.0

Double-click a row number to deactivate a variable

Variables are activated on this screen for import from external software such as Simulink. New import variables can be defined with a VS command:

```
DEFINE_IMPORT <name> = <def> ; <units> ; <desc>
```

where <name> is the new variable name that must begin with "IMP_", <def> is a default value, <units> are the units, and <desc> is a text description.

Optional commands: (14)

Figure 15. I/O Channels: Import screen.

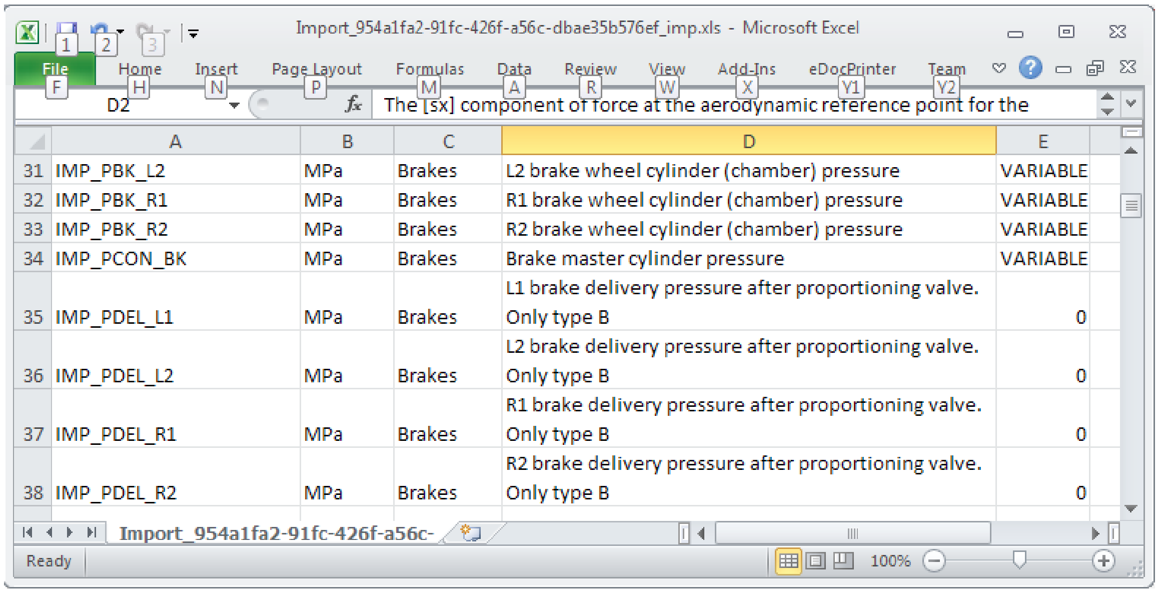
1. the dataset here must specify the same number of variables as defined in the external model, and
2. the variables must be in the same order used for the external model.

User Controls

This screen is similar to the **I/O Channels: Export** screen. This also involves selecting from available variables. However, instead of selecting from available output variables, you will select from available Import variables.

This screen also makes use of tabbed text files generated by a VS Math Model to support interactive browsing and viewing from a spreadsheet program. The controls described earlier (4 - 10) work on this screen the same as described earlier for the **I/O Channels: Write** screen. However, in this case, the tabbed text file lists Import variables, rather than output variables. As with the other screens, a corresponding CSV file can also be viewed using the **View Spreadsheet** button 7 (Figure 16).

The main difference in setting up information with this screen when compared to the **I/O Channels: Export** screen is that there are more properties to specify for each selected Import variable.


Import_954a1fa2-91fc-426f-a56c-dbae35b576ef_imp.xls - Microsoft Excel

D2 The [sx] component of force at the aerodynamic reference point for the

	A	B	C	D	E
31	IMP_PBK_L2	MPa	Brakes	L2 brake wheel cylinder (chamber) pressure	VARIABLE
32	IMP_PBK_R1	MPa	Brakes	R1 brake wheel cylinder (chamber) pressure	VARIABLE
33	IMP_PBK_R2	MPa	Brakes	R2 brake wheel cylinder (chamber) pressure	VARIABLE
34	IMP_PCON_BK	MPa	Brakes	Brake master cylinder pressure	VARIABLE
35	IMP_PDEL_L1	MPa	Brakes	L1 brake delivery pressure after proportioning valve. Only type B	0
36	IMP_PDEL_L2	MPa	Brakes	L2 brake delivery pressure after proportioning valve. Only type B	0
37	IMP_PDEL_R1	MPa	Brakes	R1 brake delivery pressure after proportioning valve. Only type B	0
38	IMP_PDEL_R2	MPa	Brakes	R2 brake delivery pressure after proportioning valve. Only type B	0

Import_954a1fa2-91fc-426f-a56c-

Ready 100%

Figure 16. View from Excel of spreadsheet with available Import variables.

As with the other screens, variables are activated by double-clicking on the name shown in the list **Available Variables** (10), which moves the variable to the list **Variables Activated for Import** (13). Some of the imported variables can be combined with internal *native* variables in the VS Math Model. If there is a native variable, then the Import can be combined with one of three modes, also specified in the table of active Imports (15). (On the other hand, if there is no native variable, then only the `ADD` mode is supported.) When a simulation starts in Simulink, the Import variables are not yet available when calculating the initial conditions. Therefore, this screen also provides fields to specify the initial values of the active Import variables (16).

(13) Names of activated Import variables. The first cell in each row has the name of an Import variable that has been activated. The order of the rows can be adjusted using the up and down arrow buttons (11). To deactivate a variable, double-click on the row number to the left of the name. (Double clicking on the name itself selects it for copying to the clipboard.)

(15) Mode for using each activated Import variable. The three possible modes are `ADD`, `MULTIPLY`, and `REPLACE`. However, not all variables support all modes.

When an Import variable is activated from the **Variables Available for Import** list, the mode control is either set to a pull-down list with all three options, or is set to display **Add** and is locked, to indicate there is no internal variable with which the activated Import can interact. This indicator is only used when the list is created or modified.

(16) Initial values for active Import variables. When the Import variables are imported from external software such as Simulink, they are given values at all times in the run except when the run starts. In some cases (e.g., Simulink), the variables are not set when the VS Math Model performs initializations. These fields are used to set the values used for initialization of the model. Once the run starts, the values are presumably set by external software.

Modes

By default, all potential Import variables are ignored unless they are activated. Within the data file read by the VS Math Model, the syntax for specifying an Import variable is:

> `[IMPORT]` *keyword* [*mode* [*initial_value*]]

where *keyword* is the name of the Import variable, *mode* is one of three possible modes for using the Import (`ADD`, `MULTIPLY`, or `REPLACE`), and *initial_value* is an expression that might no longer be useful, as noted below.

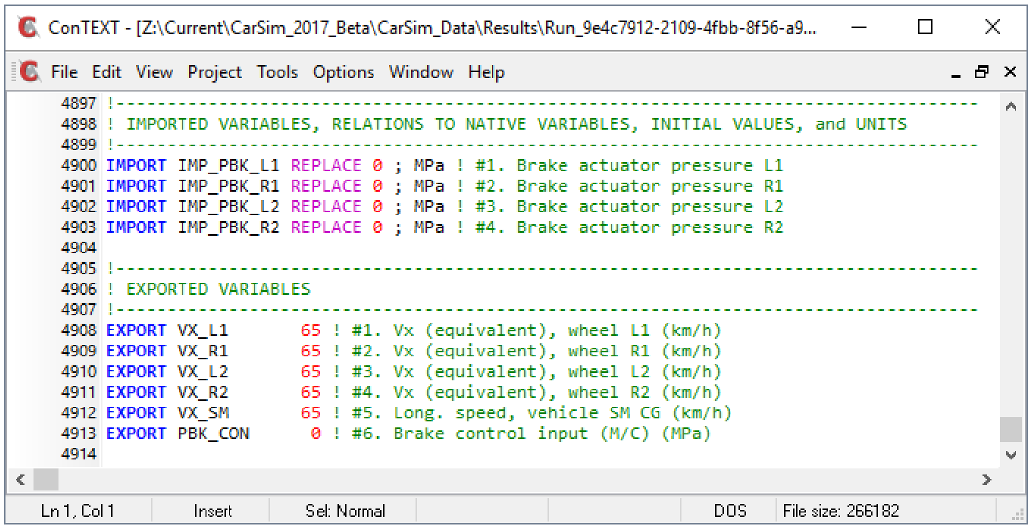
> **Note** The modes `ADD`, `MULTIPLY`, and `REPLACE` add the Import to the array that is received from external software. Three other modes are available when Imports are activated but will not be given values by external software. These are described later in the section *Activating Import Variables on Other Screens* (page 18).

Note that *mode* and *initial_value* are optional. The keyword *mode* determines how the imported variable is combined with a native value embedded in the VS model. For example, if the imported variable is the brake control and the mode is `REPLACE`, then the brake control within the VS model is replaced with the imported value. If the mode is `ADD`, then the internal value is added to the imported variable. If the mode is `MULTIPLY`, then the internal value is multiplied by the imported value. If not specified, the default is that the mode is `ADD`.

Not all Import variables support all three modes. The tabbed text and spreadsheet files indicate whether the three modes are supported, or whether there is no internal variable and the Import can only be added. This is indicated in the Import screen by the drop-down controls (15) (Figure 15); in the spreadsheet it is indicated by whether column E is `VARIABLE` or 0 (Figure 16). If 0, the mode `MULTIPLY` would have no effect (*import* × 0 = 0); in this case the modes `ADD` and `REPLACE` have the same effect of using the imported variable (*import* + 0 = *import*). Thus, variables that have 0 as the internal variable are only shown with the Add option in the drop-down control.

> **Note** Prior to version 2018.1, the array of import variables was not available at the first time step, and the optional numerical expression *initial_value* could be used to remove a transient at the simulation start. In v2018.1 the sequence in VS Math Models was changed. In newer versions, the expression *initial_value* has little or no effect in most cases (see the *VS Math Model Reference Manual* for details).

When a simulation is run using this dataset, you can look at the very end of the Echo file to confirm that the Import variables were properly set up. Figure 17 shows the end of an Echo file in which the example ABS dataset was used.


ConTEXT - [Z:\Current\CarSim_2017_Beta\CarSim_Data\Results\Run_9e4c7912-2109-4fbb-8f56-a9...
File Edit View Project Tools Options Window Help
Ln 1, Col 1
Insert
Sel: Normal
DOS
File size: 266182

```
!--------------------------------------------------------------------------------------
! IMPORTED VARIABLES, RELATIONS TO NATIVE VARIABLES, INITIAL VALUES, and UNITS
!--------------------------------------------------------------------------------------
IMPORT IMP_PBK_L1 REPLACE 0 ; MPa ! #1. Brake actuator pressure L1
IMPORT IMP_PBK_R1 REPLACE 0 ; MPa ! #2. Brake actuator pressure R1
IMPORT IMP_PBK_L2 REPLACE 0 ; MPa ! #3. Brake actuator pressure L2
IMPORT IMP_PBK_R2 REPLACE 0 ; MPa ! #4. Brake actuator pressure R2

!--------------------------------------------------------------------------------------
! EXPORTED VARIABLES
!--------------------------------------------------------------------------------------
EXPORT VX_L1       65 ! #1. Vx (equivalent), wheel L1 (km/h)
EXPORT VX_R1       65 ! #2. Vx (equivalent), wheel R1 (km/h)
EXPORT VX_L2       65 ! #3. Vx (equivalent), wheel L2 (km/h)
EXPORT VX_R2       65 ! #4. Vx (equivalent), wheel R2 (km/h)
EXPORT VX_SM       65 ! #5. Long. speed, vehicle SM CG (km/h)
EXPORT PBK_CON      0 ! #6. Brake control input (M/C) (MPa)
```

Figure 17. End of Echo file for ABS Simulink example.

Defining and Activating New Import Variables

There are occasions where you might want to add new Import variables to make use of variables from the external tool in some way. New Import variables can be using the VS Command `DEFINE_IMPORT` (see the document *VS Commands Manual*). These variables are also listed in the documentation files accessed with the **View** button on the **Run Control** screen and are activated in the same manner as the predefined Import variables.

The `DEFINE_IMPORT` commands may be placed in the miscellaneous yellow field ⑭ (Figure 15, page 15), as noted in the adjacent text on the screen. For example, Figure 18 shows an example dataset in which a new Import variable `imp_detect` is defined and added to the list: **Variables Activated for Import** ⑬.

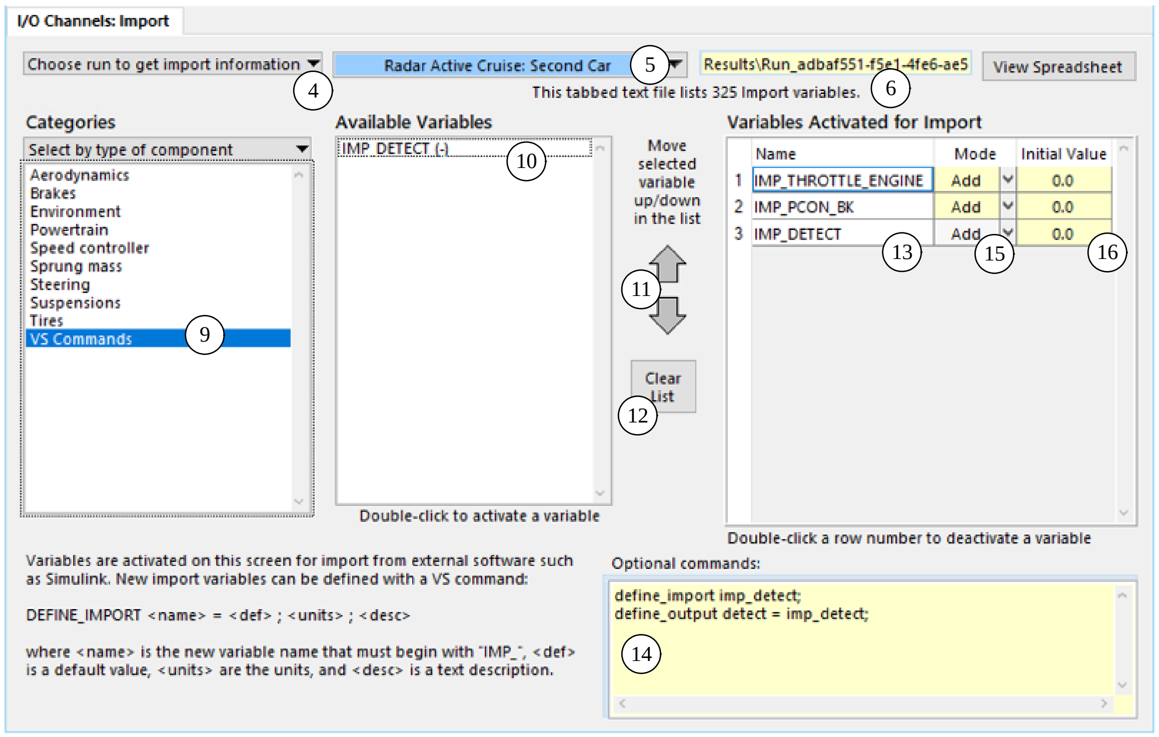
The new variable is available because the **Run Control** dataset ⑤ used to enable browsing on the screen contains a link to this dataset. The new Import is in the category **VS Commands** ⑨.

Along with the VS Command to add the new Import variable, the VS Command `DEFINE_OUTPUT` was used to define an output variable, along with an equation that sets the value each time step to the Import variable. This makes it easy to plot the imported variable with VS Visualizer along with other variables from a simulation using this dataset.

Activating Import Variables on Other Screens

The **I/O Channels Import** screen provides single point for specifying a group of Imports. When linking to external software such as Simulink, it is critical that the sequence in which they are activated matches the sequence in which they are expected in the wrapper (e.g., S-Function).

However, Import variables are not always associated with external software. For example, Figure 19 shows part of a screen used to add a set of reference points to the model, each with an associated set of forces in X, Y, and Z directions, using either the local coordinate system axes or the global axes, depending on a control on the screen ①.



Choose run to get import information: Radar Active Cruise: Second Car

Results\Run_adbaf551-f5e1-4fe6-ae5

This tabbed text file lists 325 Import variables.

Categories: Select by type of component

- Aerodynamics
- Brakes
- Environment
- Powertrain
- Speed controller
- Sprung mass
- Steering
- Suspensions
- Tires
- VS Commands

Available Variables:

- IMP_DETECT (-)

Double-click to activate a variable

Variables Activated for Import

	Name	Mode	Initial Value
1	IMP_THROTTLE_ENGINE	Add	0.0
2	IMP_PCON_BK	Add	0.0
3	IMP_DETECT	Add	0.0

Double-click a row number to deactivate a variable

Variables are activated on this screen for import from external software such as Simulink. New import variables can be defined with a VS command:

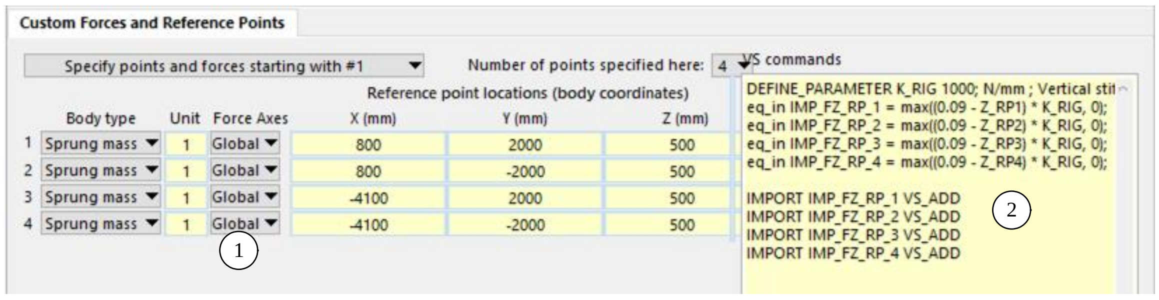
DEFINE_IMPORT <name> = <def> ; <units> ; <desc>

where <name> is the new variable name that must begin with "IMP_", <def> is a default value, <units> are the units, and <desc> is a text description.

Optional commands:

```
define_import imp_detect;
define_output detect = imp_detect;
```

Figure 18. A user-defined Import variable and an associated output variable.



Specify points and forces starting with #1

Number of points specified here: 4

Reference point locations (body coordinates)

	Body type	Unit	Force Axes	X (mm)	Y (mm)	Z (mm)
1	Sprung mass	1	Global	800	2000	500
2	Sprung mass	1	Global	800	-2000	500
3	Sprung mass	1	Global	-4100	2000	500
4	Sprung mass	1	Global	-4100	-2000	500

VS commands

```
DEFINE_PARAMETER K_RIG 1000; N/mm ; Vertical sti
eq_in IMP_FZ_RP_1 = max((0.09 - Z_RP1) * K_RIG, 0);
eq_in IMP_FZ_RP_2 = max((0.09 - Z_RP2) * K_RIG, 0);
eq_in IMP_FZ_RP_3 = max((0.09 - Z_RP3) * K_RIG, 0);
eq_in IMP_FZ_RP_4 = max((0.09 - Z_RP4) * K_RIG, 0);

IMPORT IMP_FZ_RP_1 VS_ADD
IMPORT IMP_FZ_RP_2 VS_ADD
IMPORT IMP_FZ_RP_3 VS_ADD
IMPORT IMP_FZ_RP_4 VS_ADD
```

Figure 19. Screen for creating reference points with associated forces.

The forces are added to the VS Math Model automatically. However, they all have default magnitudes of zero. Each force component has an associated Import variables that can be used to apply an action in the model.

This example is from a simulation involving limit roll behavior where test vehicles are typically equipped with outriggers. Each outrigger wheel has an associated vertical force in the global vertical direction that is generated when the wheel contacts the ground.

The miscellaneous yellow field is used to add an equation for each vertical force using the `EQ_IN` VS Command, using a stiffness `K_RIG` and the global height of the wheel center (e.g., `Z_RP1`).

The Import variables `IMP_FZ_1`, … `IMP_FZ_4` are not used unless they are activated with the `IMPORT` command, as shown (1). Notice that when they are activated, the mode is set to `VS_ADD`.

If the mode keyword has the prefix VS_ (VS_ADD, VS_MULTIPLY, or VS_REPLACE), and the VS Math Model is running with Simulink or LabVIEW, the Import variable is **not** added to the array of variables transferred from the external software to the VS Math Model.

The VS_ADD mode was specified in this example to ensure that these Imports will not conflict with Imports coming from external software, in case this dataset is attached to a vehicle that is use in a Simulink or FMI model. If the VS Math Model is run without connecting to Simulink or similar external software, then values for Import variables can only be set with VS Commands. In these cases, there is no difference between the modes with and without the VS_ prefix.

I/O Channels: Ports

The second-generation Simulink S-Function for a VehicleSim product supports Simulink models with multiple Import and Export ports. Figure 20 shows such an example, in which the CarSim S-Function has ten Import ports and twelve Export ports for various powertrain components.

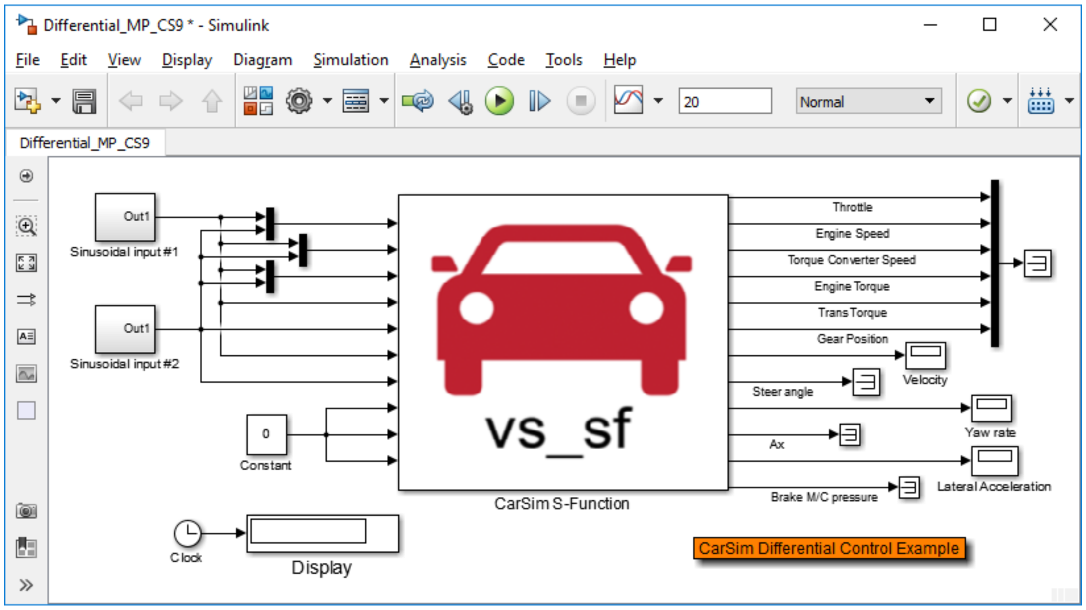
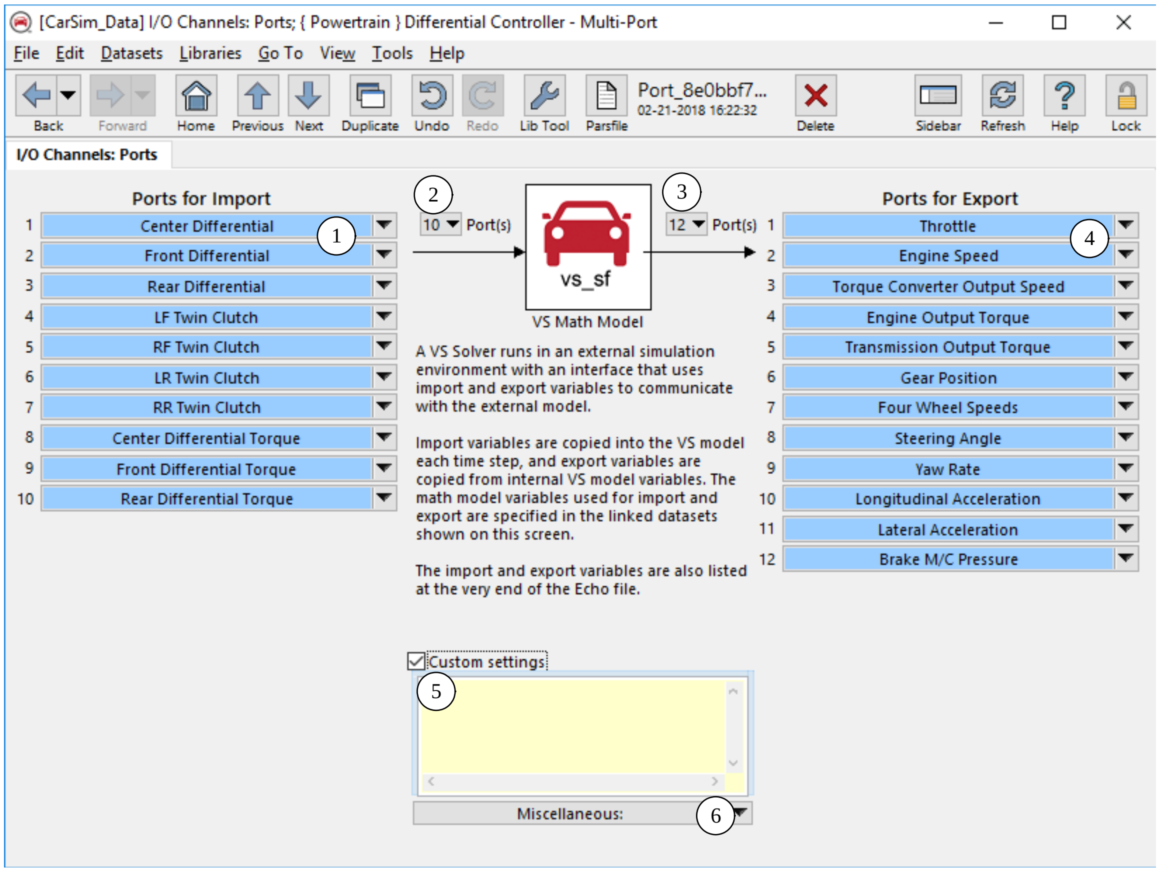


Figure 20. Simulink model with multiple Import and Export ports.

The setup from the VehicleSim side is made using the **I/O Channels: Ports** screen (Figure 21). You will usually link to a dataset from this library from the **Models: Simulink** screen.

One reason for using multiple ports is that it can be easier to combine several Simulink models used with the VS Math Model. You can copy the block diagram from one model, paste into another model, and add the Import and Export connections to the VS S-Function with new ports.

A consideration with multiple ports that does not apply for single-port models is whether it's OK to have multiple instances of Import and Export variables. For the case of Exports, there is no problem. For example, two controllers might both require the vehicle speed from the VS Math Model. If two ports are set up that both include the same Export variable Vx, two copies are made, scaled with user-defined units, and put into the array that is shared with the VS S-Function and Simulink.


[CarSim_Data] I/O Channels: Ports; { Powertrain } Differential Controller - Multi-Port
File Edit Datasets Libraries Go To View Tools Help
Back Forward Home Previous Next Duplicate Undo Redo Lib Tool Parsfile
Port_8e0bbf7...
02-21-2018 16:22:32
Delete Sidebar Refresh Help Lock
I/O Channels: Ports
1
2
3
4
5
6
VS Math Model
vs_sf

Ports for Import

1	Center Differential
2	Front Differential
3	Rear Differential
4	LF Twin Clutch
5	RF Twin Clutch
6	LR Twin Clutch
7	RR Twin Clutch
8	Center Differential Torque
9	Front Differential Torque
10	Rear Differential Torque

10 Port(s)

12 Port(s)

Ports for Export

1	Throttle
2	Engine Speed
3	Torque Converter Output Speed
4	Engine Output Torque
5	Transmission Output Torque
6	Gear Position
7	Four Wheel Speeds
8	Steering Angle
9	Yaw Rate
10	Longitudinal Acceleration
11	Lateral Acceleration
12	Brake M/C Pressure

A VS Solver runs in an external simulation environment with an interface that uses import and export variables to communicate with the external model.

Import variables are copied into the VS model each time step, and export variables are copied from internal VS model variables. The math model variables used for import and export are specified in the linked datasets shown on this screen.

The import and export variables are also listed at the very end of the Echo file.

☑ Custom settings

Miscellaneous:

Figure 21. The I/O Channels: Ports screen.

For the case of multiple instances of an Import variable, it is still possible to have multiple references to the same variable, but not recommended. For example, two controllers might both generate a brake input `IMP_PCON_BK`. The VS Math Model has only a single Import variable for that purpose, so if the Simulink model provides two different values, they cannot both be used.

If two ports both try to update the same variable, it is handled the same as when multiple datasets assign a value to a parameter: each is processed normally in the order they are sent to the VS Math Model. The last one overrides any effects of the previous ones. Thus, if an Import variable is included multiple times, the last one specified in the Simulink model is the one used in the VS Math Model.

① Data links for **I/O Channels: Imports** datasets. Each link provides the list of Import variables (Simulink signals) for the indicated port. Each visible link must be active; otherwise, Simulink will generate an error.

② **Import** drop-down control. Use the drop-down list to specify how many ports will be specified (0 to 20). A corresponding number of data links are shown ① based on this setting. This setting must match the Simulink model file (e.g., Figure 20).

③ **Export** drop-down control. Use the drop-down list to specify how many ports will be specified (0 to 20). A corresponding number of data links are shown ④ based on this setting. This setting must match the Simulink model file (e.g., Figure 20).

④ Data links for **I/O Channels: Exports** datasets. Each link provides the list of Export variables (Simulink signals) for the indicated port. Each visible link must be active; otherwise, Simulink will generate an error.

⑤ Checkbox and field for advanced users to provide miscellaneous information.

⑥ Miscellaneous link for advanced users to provide more data.