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

Windows DS for CarSim and TruckSim

Windows DS (Driving Simulator) for CarSim and TruckSim provide support for real-time operation under Windows with live animation and driver control hardware. Two approaches are provided: (1) TabletopDS (no extra software), and (2) with Simulink (from The MathWorks).

Running the examples in Windows real-time requires a Windows DS license for operation. The Simulink examples also require that Simulink be installed.

This document assumes familiarity with the operation of CarSim and/or TruckSim, including navigating the user interface (VS Browser), modifying vehicles and procedures, and linking to third-party tools such as MATLAB/Simulink.

Installing Windows DS in CarSim or TruckSim

There are two approaches for installing and activating the components of Windows DS:

1. Build a new database from the Windows DS examples and resources via the CPAR archives. This is recommended unless you need to work with existing data that you already have in other CarSim or TruckSim databases.

2. Add the Windows DS examples and resources to an existing CarSim or TruckSim database.

In either case, CarSim or TruckSim must first be installed on your computer.

Build a New Database for Windows DS

The first time you start a new version of CarSim or TruckSim, the window **Select Recent Databases** will appear (Figure 1) with an introductory message 1. (If any databases have already been built and/or converted from older versions, they will appear in a list instead of the message.)

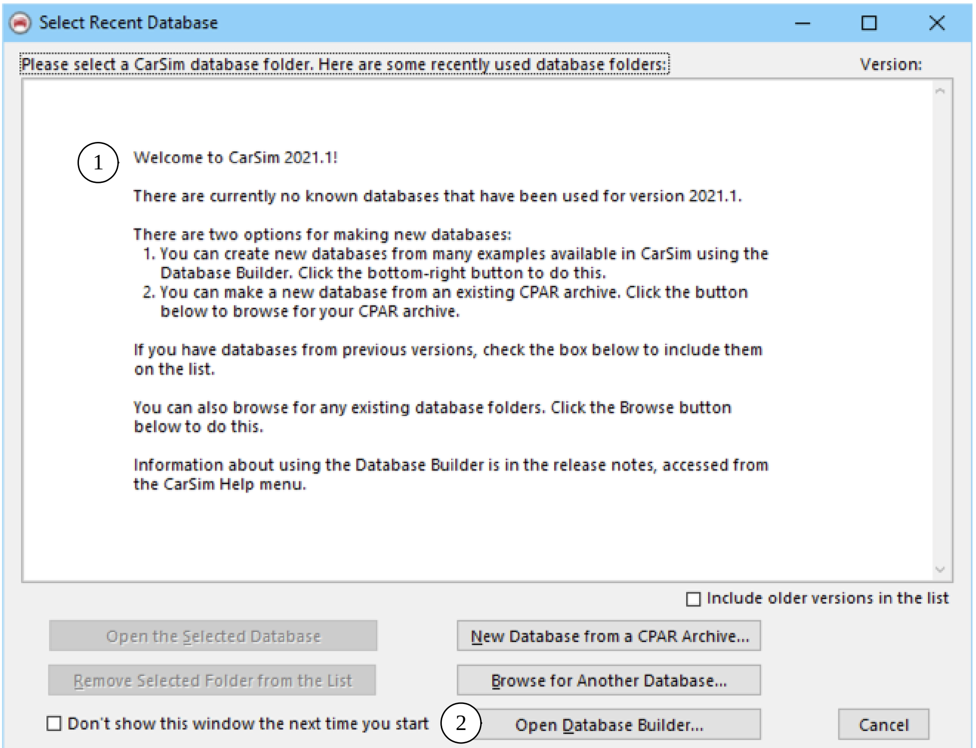


Figure 1. The Select Recent Database window the first time the software is launched.

To build a new database with the Windows DS examples, click the button **Open Database Builder** 2. Alternatively, the Database Builder may be accessed from the CarSim / TruckSim **File** menu (Figure 2).

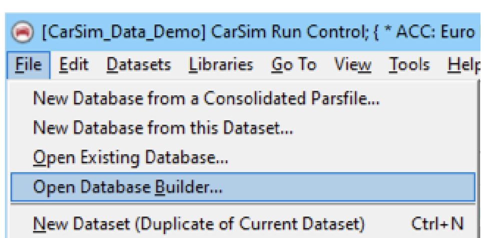


Figure 2. Open Database Builder command from the VS Browser File menu.

Figure 3 shows the Database Builder window with a selection of CPAR archives. This list consists of options of pre-defined groups of data that will be used to create a new database; when presented with this window, end-users may select anywhere from one to all the available CPAR archives to create their new database.

To make the new CarSim or TruckSim Windows DS database, follow these steps:

1. Check the box **Show all CPAR archives** 3; this will present all of the available CPAR files that can be used to create a new database.

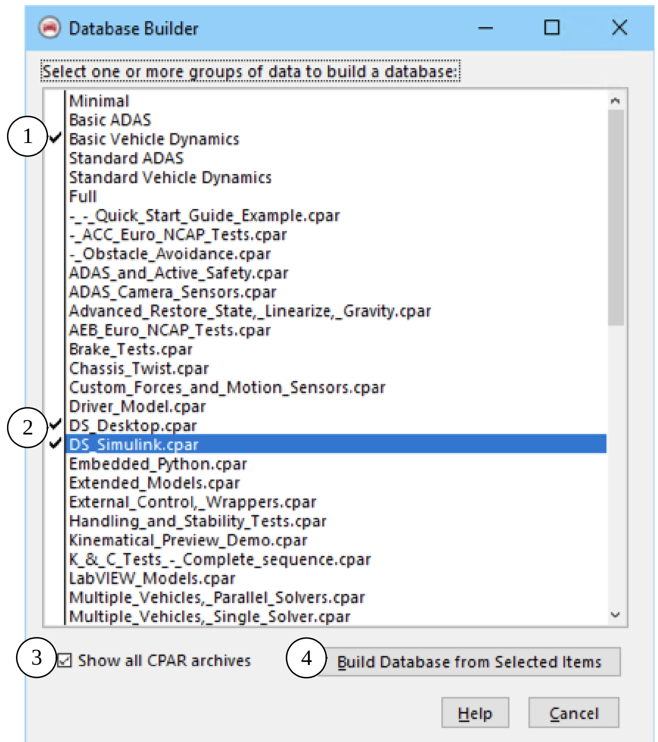


Figure 3. Use the Database Builder to build a database with DS CPAR archives.

2. Check the CPAR options for the **DS_Desktop.cpar** (no Simulink) and/or **DS_Simulink.cpar** 2. If you want the database to include more examples, check any of the groups at the top (e.g., **Basic Vehicle Dynamics** 1). Individual CPAR archives may also be selected to add specific sets of examples to be included in the new database.

3. Click the button **Build Database from Selected Items** 4.

4. CarSim or TruckSim will prompt you for the name of a new empty folder to use for the new database (Figure 4).

5. Navigate to your preferred location for a new database folder, e.g., `Documents` 1.

> **Note** It is helpful to keep CarSim and TruckSim databases grouped together by version number (e.g., all CarSim 2021.1 databases can be in a folder named `CarSim_2021_1`). This helps manage your data when you have multiple versions of CarSim or TruckSim and need to quickly locate a database.

6. Click the button **Make New Folder** 3. Windows will create a new folder with the name `New Folder` (not shown). It will initially be highlighted, serving as a prompt to provide a more descriptive name.

7. Specify a name for the new CarSim or TruckSim Windows DS database, e.g., `CarSim_Data_DS` (2).

8. After specifying the new database folder name, make sure the newly named folder is selected (Figure 4), then click the button **OK** (4).

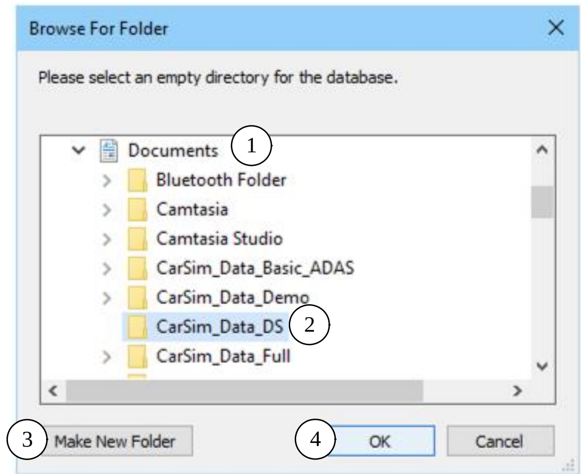


Figure 4. Prompt from CarSim to identify an empty folder for a new database.

The VS Browser will now extract information from the selected CPAR(s) and create the new database. The time needed to do this depends on the size of the selected CPAR file(s). If only the Windows DS examples are selected, this takes about 15 seconds. If other CPAR files are included, the process can take several minutes. Once the new database has been built, all datasets will be locked. Use the **Tools** drop-down menu to select **Unlock Everything in the Database**.

Add Windows DS Examples to an Existing Database

An alternative approach is to add the Windows DS examples to an existing database. To do this, follow these steps.

1. Launch CarSim or TruckSim and select the database to which the Windows DS examples will be added.

2. From the **File** drop-down menu (Figure 5, (1)), select the option **Import Parsfile (Any Export Type)** (2).

3. You will be prompted to select a file to import (Figure 6). The CPAR archives are in the Prog folder for your product (e.g., `CarSim_Prog` (1)), in the subdirectory `Resources\CPAR_Archives` ((2) and (3)). As noted previously, there are two CPAR archives for DS: `DS_Desktop.cpar` (Simulink not used) and `DS_Simulink.cpar`. Choose one of these (4) and click the **Open** button (5).

> **Note** If you want to import both CPAR files, you will need to repeat the **Import** process for the second one.

4. You will be prompted for settings to handle possible duplicate datasets (Figure 7). Check the boxes shown in the figure (1), then click the button **OK** (2).

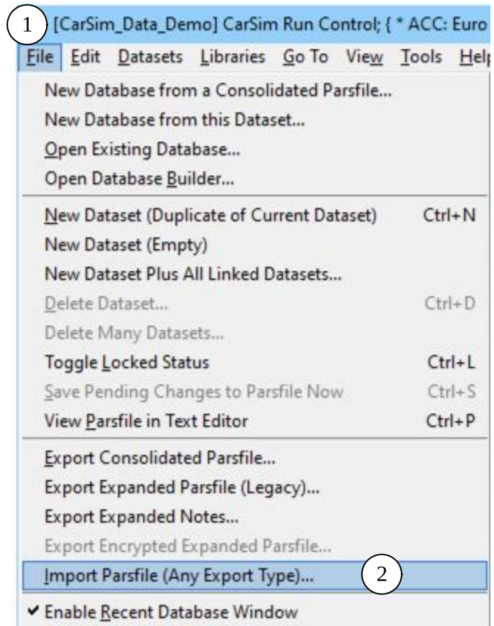


Figure 5. Select Import Parsfile (Any Export Type) from the File menu.

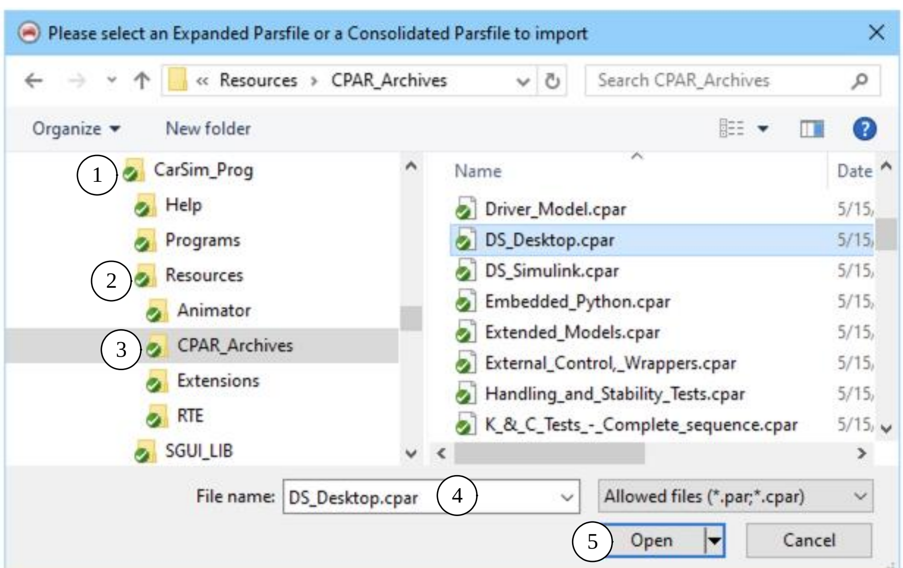


Figure 6. Select a DS CPAR archive to import.

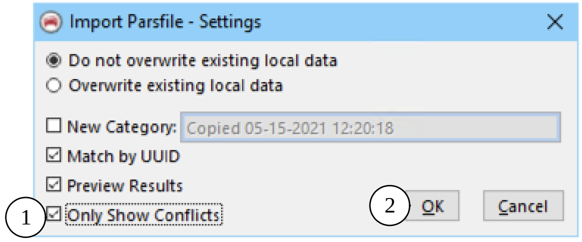


Figure 7. Import settings to handle duplicate datasets.

5. The browser will show a window summarizing the data to be imported (Figure 8). Ideally, the **Import Action** column ① will only show the action **Skip File** ② (meaning a dataset already exists in the database, therefore the corresponding file in the CPAR will not be imported). To check for any possible conflicts, either scroll down the list or click the column title (**Import Action** ①). This will reverse the order of the rows. If all the rows still show **Skip File**, no parsfile conflicts exist and you can continue with the parsfile import operation by clicking the button **OK** ③.

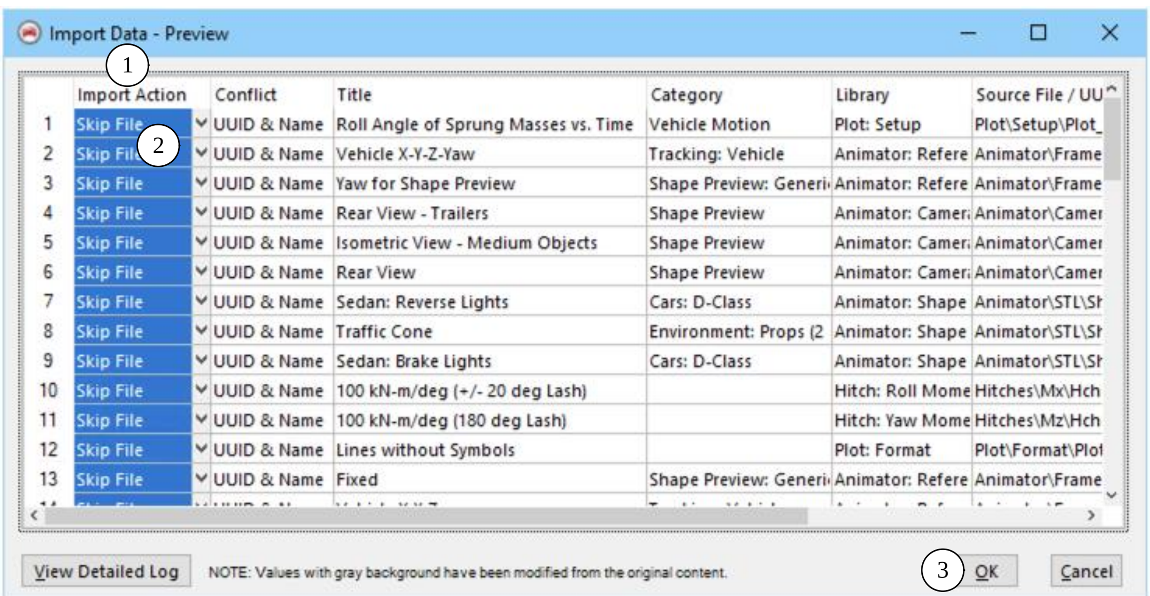


Figure 8. Preview window for imported data, showing potential conflicts and action.

Note If the preview window shows any conflicts (usually with a modified title shown with a gray background), please see the *VS Brower Reference Manual* for options.

When exporting parsfiles from one database and importing them into another, the possibility for conflicts always exists. Sometimes it is as simple as the same dataset exists in both the CPAR and the database into which the CPAR will be imported, and both datasets contain the same data. Other times, the issue could be two datasets with the same name, but each contain different data. When in doubt as to which import options to select, it is best to use the CPAR to create a new database, then check the contents of the parsfiles in question with those in the database to which you want to import into. From there, you can decide if any potential conflicts are acceptable; if not, you can decide whether they should be addressed before importing the CPAR.

Installing the Logitech Hardware: G27, G29, and G920

Windows DS for CarSim and TruckSim support three sets of Logitech hardware: G27, G29, and G920.

For the G27, the wheel-mounted paddles and the H-pattern shifter can be used for both automatic and manual transmission vehicles. An exception is in TruckSim, where the H-pattern shifter is used in conjunction with a sequentially shifted manual transmission.

For the G29 and G920, Windows DS examples make use of the wheel-mounted paddles for both automatic and manual transmission vehicles, while the optional H-pattern shifter is used only for vehicles equipped with a manual transmission.

The use of the Logitech wheels starts with the installation of the Logitech drivers. Logging into Windows with administrator permission is typically required.

1. Install the Logitech hardware and software according to the manufacturer's instructions. They can be obtained from the Logitech website, or installation materials packaged with the device.
2. The following Logitech wheel settings should be used with Windows DS (more details below):
    a. Rotation value of 900°
    b. Single-axis wheel

The Logitech Drivers for the G27 and G29/G920 have not historically been compatible with each other. Troubleshooting steps regarding driver compatibility and usage of the Logitech Gaming Software with the G29 and G920 can be found in the *Troubleshooting* section at the end of this document.

Note Although Logitech no longer sells the G27, the hardware is still supported by Mechanical Simulation. If Logitech discontinues support for the G27, Mechanical Simulation cannot guarantee support for this hardware in future Windows DS releases.

Logitech Steering Wheel Settings

Successful operation of the Logitech wheel, pedals, and shifter with Windows DS requires modification to the default settings from the Logitech installation. Please follow these steps.

1. With the Logitech hardware plugged into your computer, use the Windows **Control Panel** and select **Devices and Printers.**

2. From **Devices and Printers**, right-click the Logitech hardware (e.g., G27 Racing Wheel) and select **Game controller settings**.
3. From the **Game Controllers** panel, click **Properties** (Figure 9, 1).

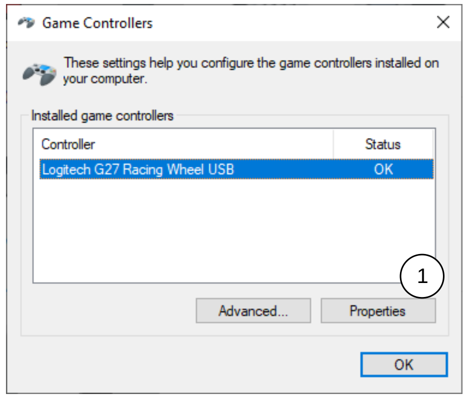


Figure 9. Logitech Game Controllers panel.

4. On the **Logitech Racing Wheel Properties** panel, click **Settings** (Figure 10, 1).

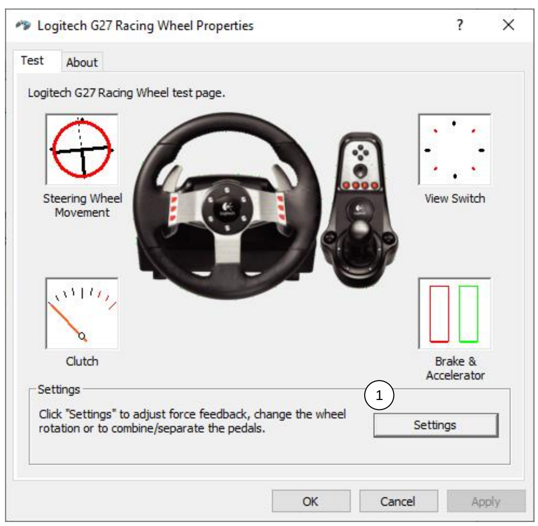


Figure 10. Logitech Racing Wheel Properties panel.

5. On the **Settings** panel (Figure 11), make the following changes:

   1 Uncheck **Combined (single axis – used for most games).**

   2 Set **Degrees of Rotation** to 900^{0}.

   3 Check the box **Enable Force Feedback**.

6. Once the modifications are complete, click **Close** to close the **Settings** panel.
7. Upon returning to the **Logitech Racing Wheel Properties** panel, you may test the various functions of the wheel, pedals, and shifter.

8. Once you are satisfied with the operation of the hardware, click **OK** to close the panel.

Setup of the Logitech hardware has been completed, and we are now ready to use the Windows DS examples.

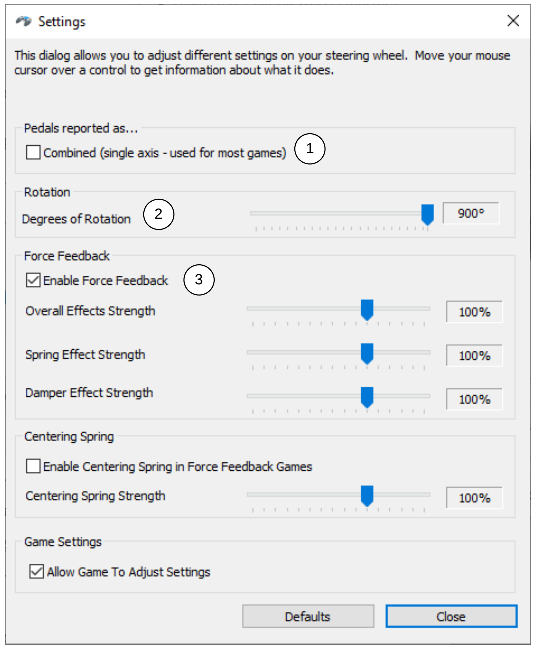


Figure 11. Logitech Settings panel.

Operation of Windows DS for CarSim and TruckSim

When running Windows DS, the VS Math Model imports driver control signals from the connected hardware (e.g., Logitech G29) and exports steering wheel reaction torque. This gets applied by the servo in the steering wheel, representing the force-feedback used by the Logitech hardware.

The Windows DS databases include **Run Control** examples organized into the two categories **DS: Desktop** and **DS: Simulink** (runs from each category were either installed or imported from a corresponding CPAR archive). In each category, there are multiple runs covering different vehicles, environments, and Logitech controllers.

In all cases you will start a simulation from the **Run Control** screen, resulting in the following:

1. VS Visualizer (VSV) is launched in a Live Video mode.

2. The control software for Windows DS is launched (i.e., either Tabletop Driving Simulator or Simulink).

Both the control software and VSV can remain open when stopping and then restarting a simulation, saving time when making changes to the vehicle and evaluating the results run-to-run. However, if a new environment is selected — e.g., switching from the virtual proving ground to Mcity — the intended operation is that VSV is closed, the new environment is selected from CarSim or TruckSim, and then VSV is relaunched for the new simulation. This action allows VSV to load the animator shapes and sound files associated with the new environment.

If VSV does not launch as expected, there are several possible reasons why. The most common are:

1. The number of live server connections is not set to 1 on the **Run Control** screen.
2. The "Extra Live Animator" license option is not checked on the **License Settings** window.
3. On the **Models** screen, the keyword `LIVE_SERVER` is not set to localhost.

Please see the *Troubleshooting* section at the end of this document for more details.

Tabletop Driving Simulator

Tabletop Driving Simulator is a wrapper program that runs CarSim and TruckSim VS Math Models in Windows real-time mode. Tabletop DS handles communication between the VS Math Model, the Logitech hardware, and VSV. It also has GUI displays showing information about the run.

Run Control screen

Figure 12 shows the top part of the **Run Control** screen set up to work with the Tabletop Driving Simulator. The circled numbers in the figure identify some settings that are specific for a DS Desktop setup.

1 The linked **Math Model** vehicle dataset has been checked for use in the Windows DS examples for a range of dynamic conditions that might be encountered.

2 The linked **Procedure** dataset disables the built-in driver control options. Instead of using the built-in driver controls, the simulation relies on controls imported into CarSim and TruckSim from the external hardware. It also sets up most of the environment (e.g., roads and paths) and includes many plot settings for variables that might be of interest in post-processing visualization.

3 The linked Miscellaneous dataset has settings to generate head-up-displays (HUD) for variables that might be of interest. These appear in the upper-left corner of the driver view in VSV (Figure 13).

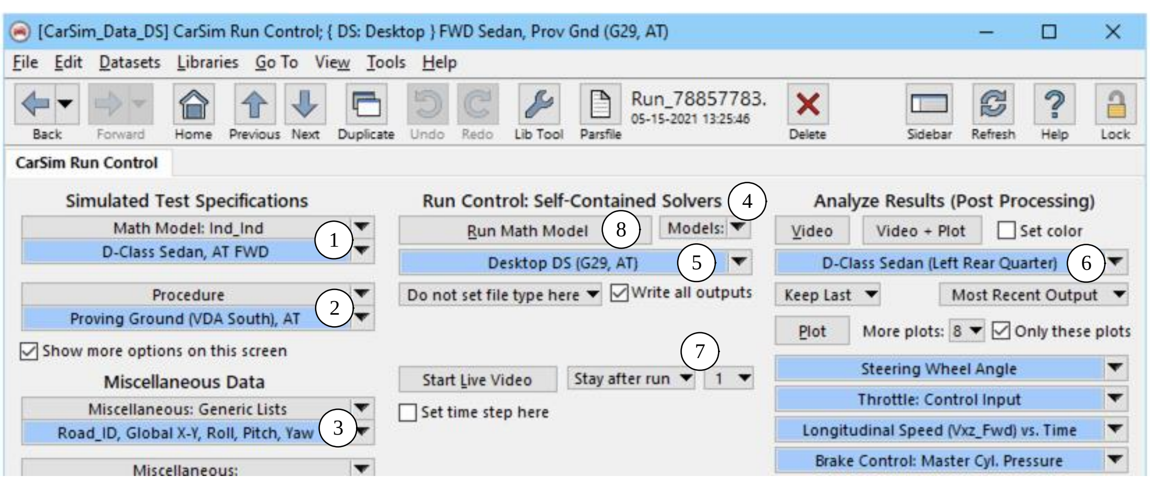


Figure 12. Run Control screen appearance for a simulation using DS Desktop.

Figure 13. Driver view of the Windows DS proving ground.

④ The **Models** drop-down control is used to specify the library **Self-Contained Solvers**. This supports the use of the external wrapper program `TabletopDS.exe`.

⑤ Link to a dataset in the **Models: Self-Contained Solvers** library, shown in the next subsection.

⑥ Link to an **Animator: Camera Setup** dataset. There are typically two choices: one for the driver view (Figure 13), and another for an external view that is helpful for post-processing visualization. The same linked **Animator: Camera Setup** dataset is used for both the Live Video and when clicking **Video** or **Video + Plot** for post-processing. If a different camera view is desired for post-processing, that selection can be made here.

⑦ Settings to control the live animation used during the Windows DS simulation.

⑧ **Run Math Model** button, used to launch the TabletopDS wrapper program.

Models: Self-Contained Solvers screen

Figure 14 shows the **Models: Self-Contained Solvers** screen for the **Run Control** dataset shown in Figure 12, with control settings typical for most Windows DS setups.

① Checkbox to show the field for an external wrapper program ②.

② Field to specify an external wrapper program, e.g., `TabletopDS_G29_AT.exe`.

③ Data field to specify the **Live animator refresh rate** for VSV (keyword = `LIVE_SERVER_UPDATE_FREQ_MAX`).

④ Checkbox to specify the numerical integration method and math model settings for the simulation. These settings will override the global settings for this database, set on the **Preferences** screen (**Tools** > **Preferences**).

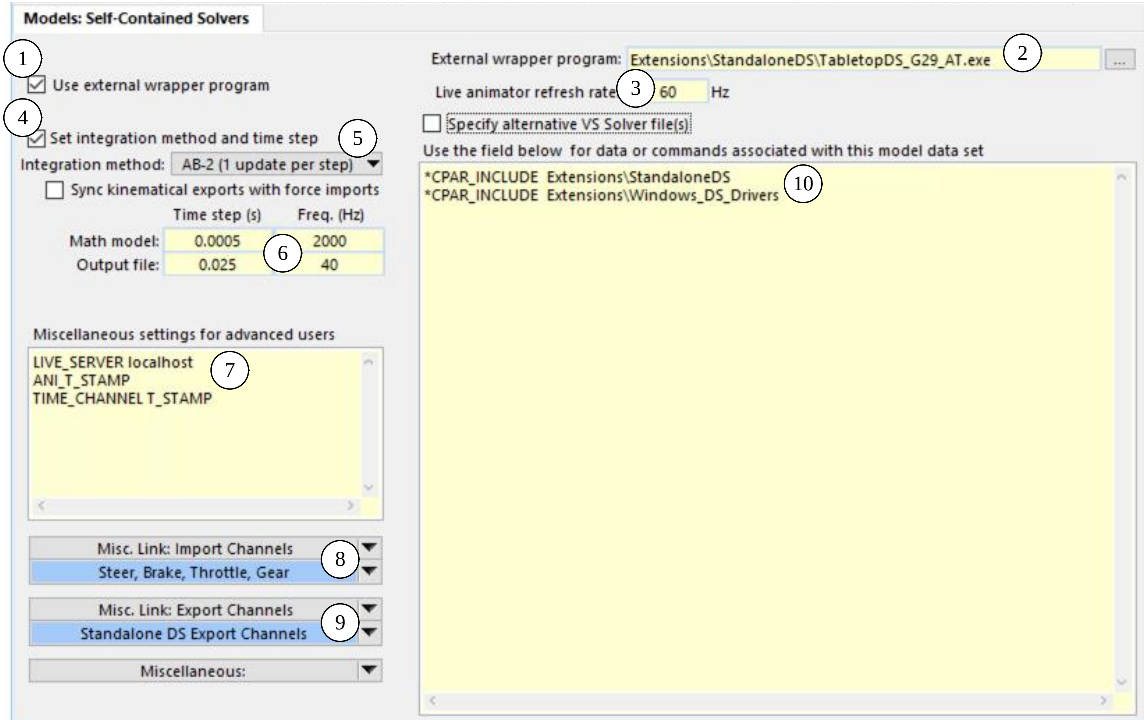


Figure 14. Models screen appearance, after clicking the dataset 5 in Figure 12.

5 Drop-down menu to select the numerical integration method. CarSim and TruckSim support six numerical integration methods with either one or two updates per time step. The numerical integration method determines how often the external wrapper communicates with the VS Math Model time step settings 6.

The numerical integration methods are described in a Technical Memo, available from **Help** > **Technical Memos** > **Numerical Integration in VS Solvers**.

6 Data fields for the internal math model time step and for recording to the simulation results file. The time step can be specified using either seconds or frequency, and the VS Browser automatically maintains the correct inverse relationship between these two values: If you modify the time step, the frequency is calculated and updated. Likewise, if you modify the frequency, the time step is calculated and updated. This guarantees that these numbers are multiples of the internal time step used by the VS Solver.

More information regarding the settings for math model time step, output file size, and their relationship to the VS Solver can be found in the **Models** documentation (click the **Help** icon in the upper right corner of the current **Models** screen).

7 **Miscellaneous** yellow field. Used in this example to set Live Animation control details.

8 Link to an **I/O Channels: Import** dataset, used to define the list of Import Variables that will be read by the VS Solver during the simulation. This list determines the array of data the VS Math Model will receive each time step from the Logitech hardware.

9 Link to an **I/O Channels: Export** dataset, used to define the list of Export Variables that will be sent to the external wrapper program and ultimately the Logitech hardware. These variables

are calculated by the VS Math Model and can include user-defined equations written with VS Commands.

10 **Miscellaneous** yellow field. Used in this example to specify that the folders `Extensions\StandaloneDS` and `Extensions\Windows_DS_Drivers` will be included in any CPAR archive that includes this **Models** dataset (e.g., **File** > **Export Consolidated Parsfile**…). The folder `StandaloneDS` contains the various wrapper programs (i.e., `TabletopDS.exe`) for use with the various Logitech hardware. The other folder — `Windows_DS_Drivers` — contains the Logitech drivers. Both folders are necessary to ensure successful operation of Windows DS.

I/O Channels: Import screen

The **I/O Channels: Import** screen allows the end-user to define the Import Array, a list of variables that represent data to be brought into the VS Solver every time-step and evaluated as part of the overall CarSim and TruckSim simulations. In the case of Windows DS, the import variables (Figure 15, 2) represent the controls from the Logitech hardware. The type of import mode supported by the Import Variable of interest — most variables support Add, Replace, or Multiply, while others only support Add — determines how the corresponding control is setup within the VS Browser (typically on the **Procedure** screen).

The Windows DS examples are pre-configured to run with the Logitech wheel, pedals, and shifter and require no modification by the end-user. If a different combination of import variables is desired, the following options are available:

1 **Help** button for this screen. Click to bring up the documentation describing the steps needed to set up new datasets.

2 Import Variables that define the Import Array. For the Windows DS examples, these are: Steering Wheel Angle, Master Cylinder Brake Pressure, Throttle, Clutch Control (manual transmission), Transmission gear, and the Transmission mode (Reverse, Neutral, 1st for Open Loop Shifting, or 2 – 18 for Closed Loop Shifting). (see page 21 for HEV and EV vehicles).

I/O Channels: Export screen

The **I/O Channels: Export** screen allows the end-user to define the Export Array, a list of variables that are calculated by the VS Solver every time-step and supplied to `TabletopDS.exe`, Simulink, or VS Commands. In the case of Windows DS, the Export Variables (Figure 16, 2) represent vehicle data provided to the control software (`TabletopDS.exe` or Simulink) and the Logitech hardware. As with the **I/O Channels: Import** screen, there is on-line **Help** (Figure 16, 1) describing the operation of the GUI options for activating any of the available Import variables.

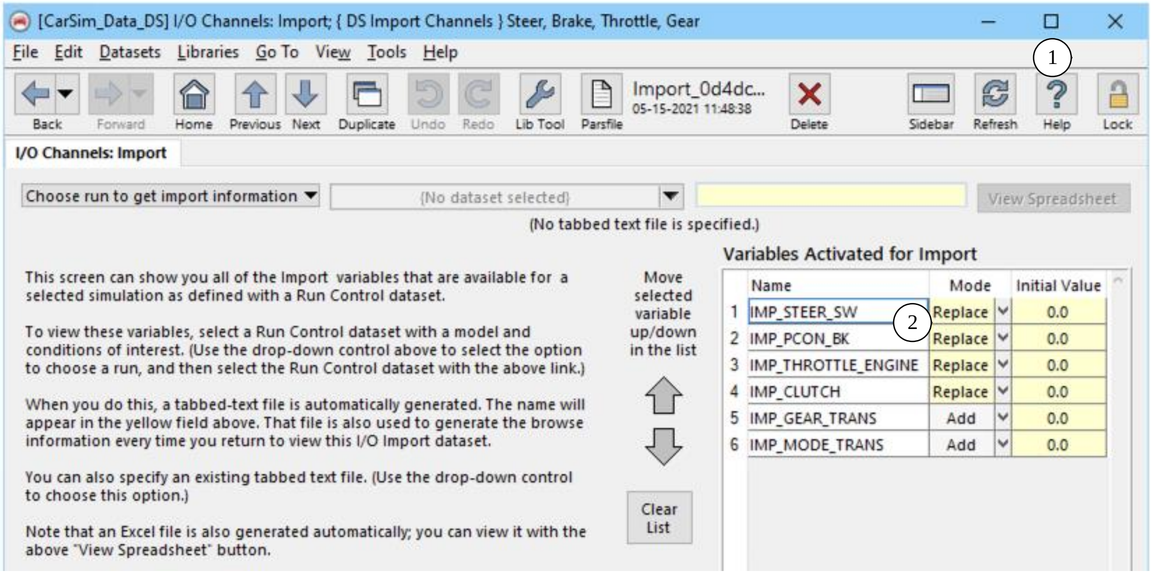


Figure 15. I/O Channels: Import screen, with specification for six import variables.

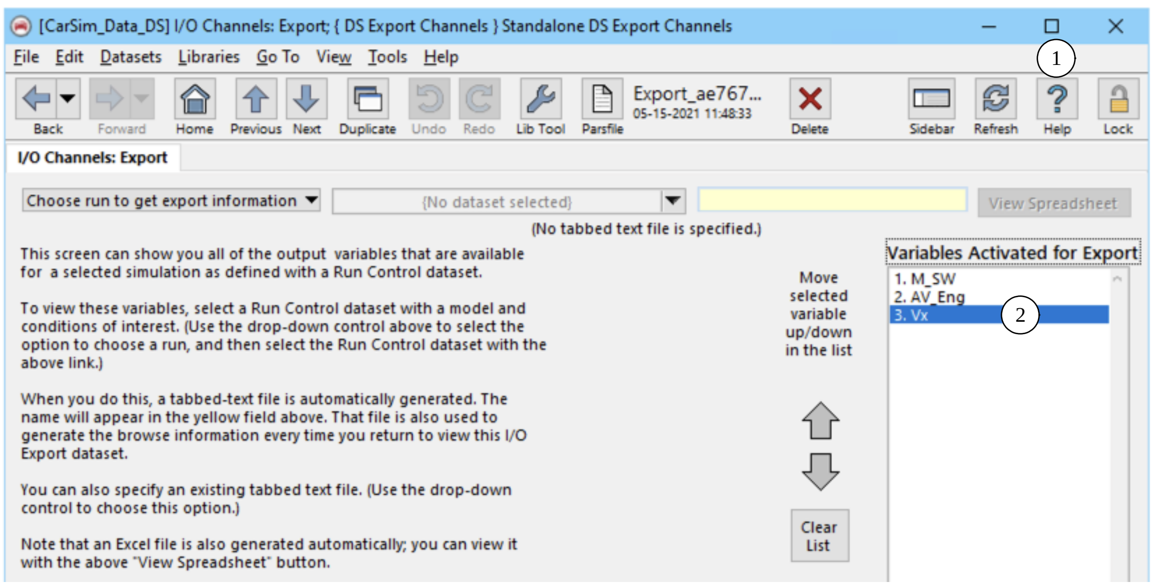


Figure 16. I/O Channels: Export screen, with specification for three export variables.

Start driving a simulation

If the **Run Control screen** (Figure 12, page 10) is locked, unlock it (**Tools > Unlock Everything in the Database**). Click the **Run Math Model** button 8 to launch VSV and the Tabletop Driving Simulator wrapper program.

The Tabletop Driving Simulator GUI (Figure 17) has a variety of vehicle and simulation information that is updated during the simulation including a speedometer, tachometer, bar gauges for the brake and throttle pedal positions, as well as fields reporting the simulation's start and stop times, math model time step, current simulation time, steering wheel angle, and the transmission gear position.

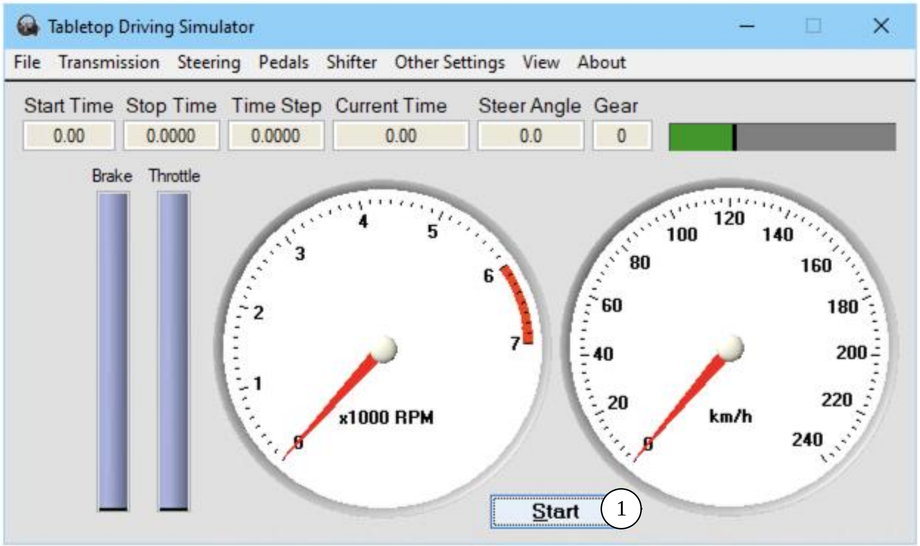


Figure 17. Tabletop Driving Simulator window.

Start the simulation by clicking the **Start** button ①. When the simulation is running, the button title changes to **Stop**, allowing the same button to be used to stop the simulation.

Adjust settings

The Tabletop Driving Simulator interface has a variety of drop-down menus used to adjust settings related to the hardware and driver controls. Figure 18 and Figure 19 illustrate a few of these options.

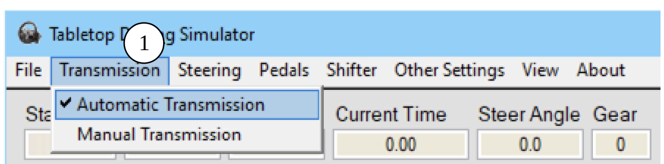


Figure 18. Drop-down menu for Transmission (Automatic or Manual).

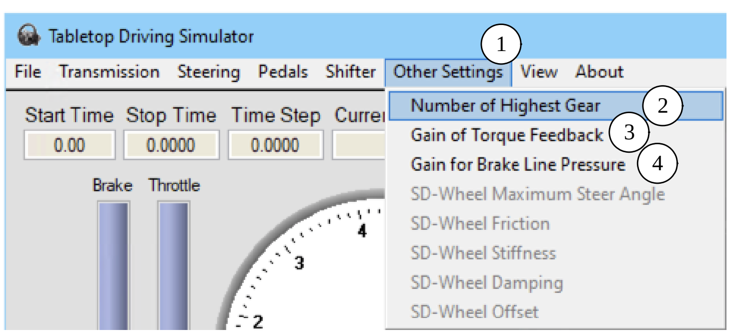


Figure 19. Drop-down menu with options to adjust driver controls.

The most common settings that are adjusted for a simulation with the Tabletop Driving Simulator are accessed from the drop-down menu **Other Settings** (Figure 19, ①).

2 Number of Highest Gear. CarSim and TruckSim have two settings used to determine the number of transmission gears available in a simulation: Number of transmission gears, set on the powertrain's **Transmission** screen (up to 18 forward gears); and the maximum gear the transmission is allowed to shift up to. For Windows DS examples, this is set to 18 (Figure 20, 2).

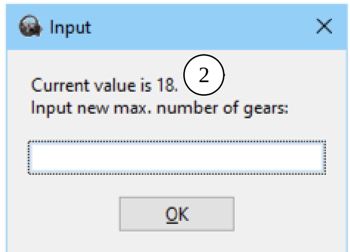


Figure 20. Window to set a new value.

The max gear setting of 18 does not require that the vehicle's transmission have 18 forward gears. To ensure more flexibility and robustness with various vehicles when using the `Tabletop.exe` application with Windows DS, it is convenient to set the maximum value here to 18. From the standpoint of CarSim and TruckSim, this setting will be stored by the math model. If the vehicle's transmission has 18 forward gears and the maneuver and vehicle data allow the transmission to reach 18th gear, this setting will allow that. If instead the transmission only has 10 forward gears, then that is the highest gear the transmission will reach. If instead we were to set the maximum value here to 10 and then select a vehicle with 18 forward gears, the maximum gear the transmission could reach would be 10, regardless of the maneuver. Therefore, by setting this to 18, this ensures it can be used with all vehicles in CarSim and TruckSim that use a geared transmission.

3 Gain of Torque Feedback. This represents a gain on the steering wheel reaction torque calculated by the VS Solver and sent to the Logitech hardware. Increasing the gain will produce more reaction torque with the trade-off being the wheel is more sensitive to abrupt changes in the tires' forces and moments (e.g., contacting a bump, engaging the ABS). Decreasing the gain reduces this sensitivity and the wheel's force feedback. The effect is that the wheel can feel disconnected from the car, especially on low mu road surfaces.

Common values are in the range of 0.2 to 0.25, however experimentation may be necessary to identify the ideal value for your simulation. Keep in mind that tire force and moment properties, as well as steering system and suspension data, all influence the calculated handwheel reaction torque from the VS Solver.

4 Gain for Brake Line Pressure. Similar to the Gain of Torque Feedback (3), this represents a gain on the Logitech hardware's brake pedal, corresponding to the vehicle's Master Cylinder Brake Pressure. Increasing the gain makes the brake pedal more sensitive and results in an increased amount of brake pressure calculated by the VS Solver for a given amount of physical brake pedal travel. Decreasing the gain reduces the sensitivity to pedal travel and consequently the amount of brake pressure sent to the brake system.

A value of 9 corresponds to 9 MPa being imported into the VS Solver; this can be confirmed by looking at a plot of Master Cylinder Pressure from within CarSim or TruckSim.

The window has two display modes. To switch to the Compact View, use the **View** menu (Figure 21, 5). The compact view (Figure 22, 6) has a button to switch back to the Full View.

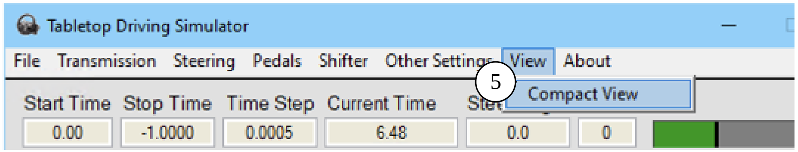


Figure 21. Use the View menu to switch to the Compact View mode.

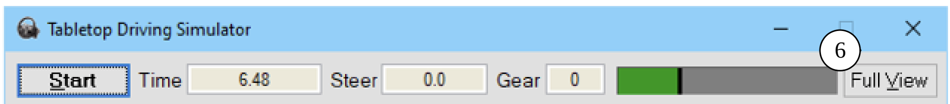


Figure 22. Use the Full View button to switch back to the Full View.

Saving settings

The Tabletop Driving Simulator wrapper is in the CarSim or TruckSim database folder `Extensions\StandaloneDS` (Figure 23, 1). In this example, the wrapper specified in Figure 14 was `TabletopDS_G29_AT.exe` 3. The wrapper saves the configuration settings in a file with the same name as the wrapper `.exe` file, but with the extension `.cfg`. For example, the configuration for `TabletopDS_G29_AT.exe` 3 is saved in the corresponding file `TabletopDS_G29_AT.cfg` 2.

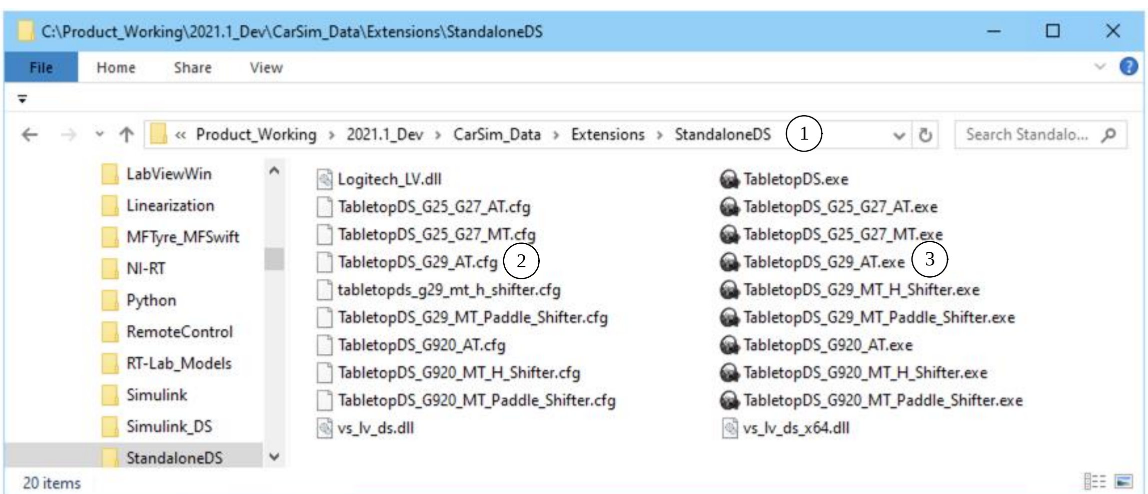


Figure 23. StandaloneDS folder, containing TabletopDS.exe and matching Configuration files.

The `.cfg` file is rewritten each time the `.exe` file quits. To make a new version with different settings, the process is:

1. Copy the file `TabletopDS.exe`
2. Rename the copy to something useful (e.g., `TabletopDS_June_2021.exe`).

3. Launch the copy and make new settings in the Tabletop Driving Simulator GUI (Figure 17).
4. Exit the new TabletopDS program. Notice that a new `.cfg` file is created in the same folder with the same name as the new `.exe`.
5. To use the new settings, specify the new TabletopDS executable (`.exe`) in the **Models: Self-Contained Solvers** dataset (Figure 14, page 12, 2).
6. Every time this new `TabletopDS.exe` is executed, the settings are saved to the new `.cfg` file with the same root name as the new `.exe` created as part of these steps.

Running with Simulink

Simulations involving Windows DS and Simulink use a VS S-Function and a corresponding Logitech block in a Simulink model to share Import and Export variables. The block **RT Enable** slows the Simulink model down to run in Windows real-time.

Controls for adjusting the model configuration — Gain of Torque Feedback, Maximum number of transmission gears, Gain for Master Cylinder Brake Pressure, etc. — are represented as standard Simulink gain blocks. Different Simulink models are used, based on the Logitech hardware and whether the vehicle has an automatic or manual transmission.

One advantage of running Windows DS with Simulink is the option to incorporate additional MATLAB and Simulink-based controls into the simulation.

Run Control screen

The **Run Control** screen layout and controls for a Windows DS simulation with Simulink (Figure 24) are nearly identical to those with Tabletop Driving Simulator (Figure 12, page 10). The only differences are that the **Models** link 4 is made to dataset in the **Models: Simulink** library 5, and the **Run Math Model** button is replaced with the buttons **Run Now** and **Send to Simulink** 8.

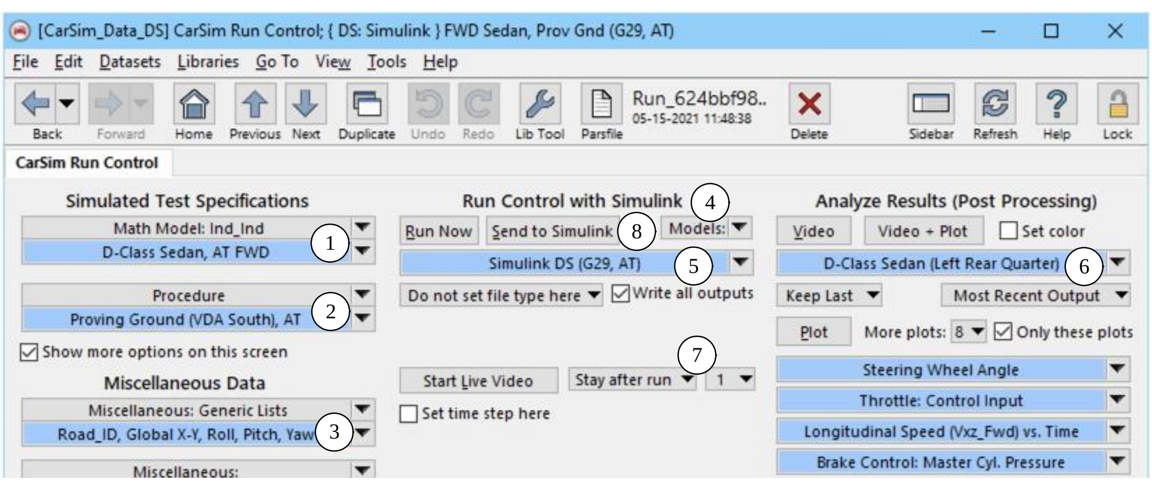


Figure 24. Run Control screen appearance for a simulation using Simulink.

(5) Link to a dataset in the **Models: Simulink** library, used to specify the Simulink model, Import and Export Variables, VS Commands for Live Animation control, numerical integration options, math model time step size, and corresponding output file size.

(8) **Send to Simulink** button, used to launch the Simulink model and VSV.

> **Note** Do not use the button **Run Now** for Windows DS. Clicking **Run Now** launches Simulink, runs the model, saves it, and closes it. Although appropriate for running many consecutive simulations (e.g., as in a DOE), using this with Windows DS prevents proper control via the Logitech hardware.

Models: Simulink screen

Click on the blue link for the **Models: Simulink** library (Figure 24, (5)) to view the dataset **Simulink DS (G29, AT)** (Figure 25). Controls on this **Models** screen are nearly identical to the **Models: Self-Contained Solvers** datasets shown earlier (Figure 14, page 12); the main differences are a link to the Simulink model and potentially a working directory that is outside of the CarSim or TruckSim database ((1), (2)).

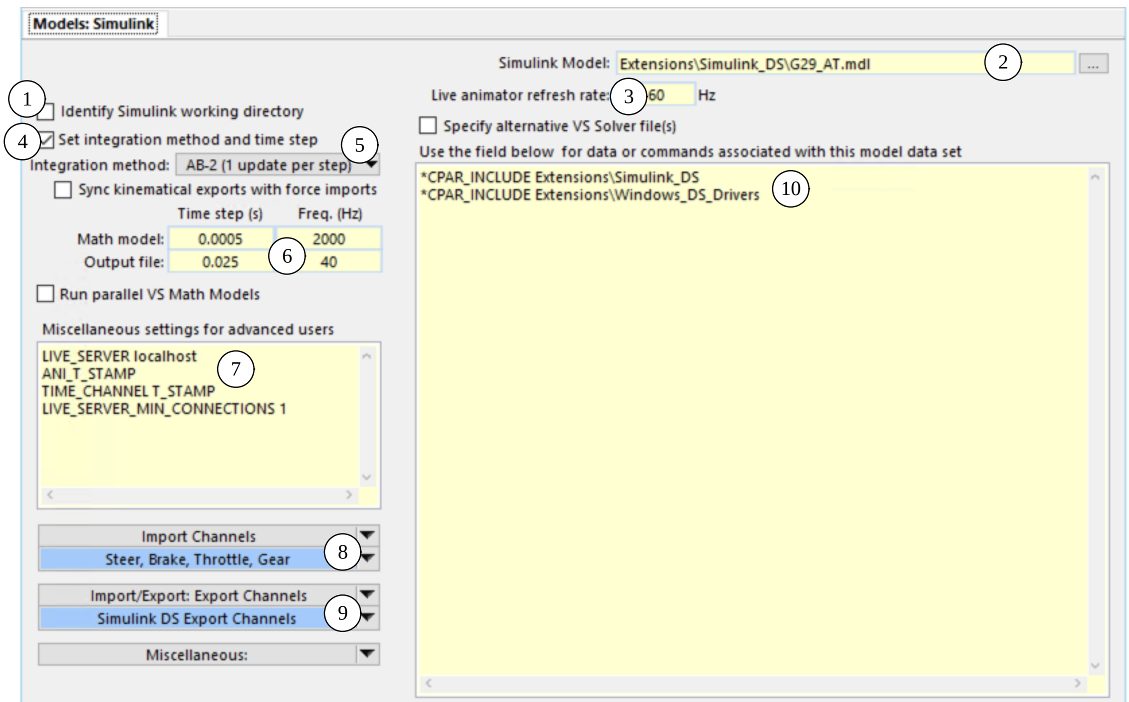


Figure 25. Models: Simulink screen after clicking the dataset (5) in Figure 24.

Start the simulation

From the **Run Control** screen (Figure 24), click the **Send to Simulink** button ((8)) to launch VSV and the Simulink model specified on the **Models: Simulink** screen (Figure 25, Figure 26).

(1) Simulink **Start** button. Clicking this button starts the Simulink model and with it the Windows DS simulation. Once the Simulink model is running, this button changes to a **Stop** control in the same manner described earlier with Tabletop Driving Simulator (Figure 17, (1)).

(2) Simulink mask and S-Function for the Logitech hardware. The S-Function allows the Logitech hardware to communicate with the Simulink model and the VS Math Model. Inputs to this block are the VS Math Model-calculated steering wheel torque and the transmission gear position selected from the Logitech steering wheel mounted paddles or H-pattern shifter. Outputs to the Simulink model represent the steering wheel angle, throttle position, brake pedal position (converted to a pressure), transmission drive mode (Reverse, Neutral, or 1st gear, etc.), and clutch pedal position (manual transmission vehicles only).

(3) **RT Enable** block, required to slow down the Simulink model to allow it to run in Windows real-time.

(4) Various gains on the driver controls, functionally equivalent to the **Other Settings** drop-down menu options used in Tabletop Driving Simulator (Figure 19, page 15, (1) - (4)). These gains are adjustable by the end-user.

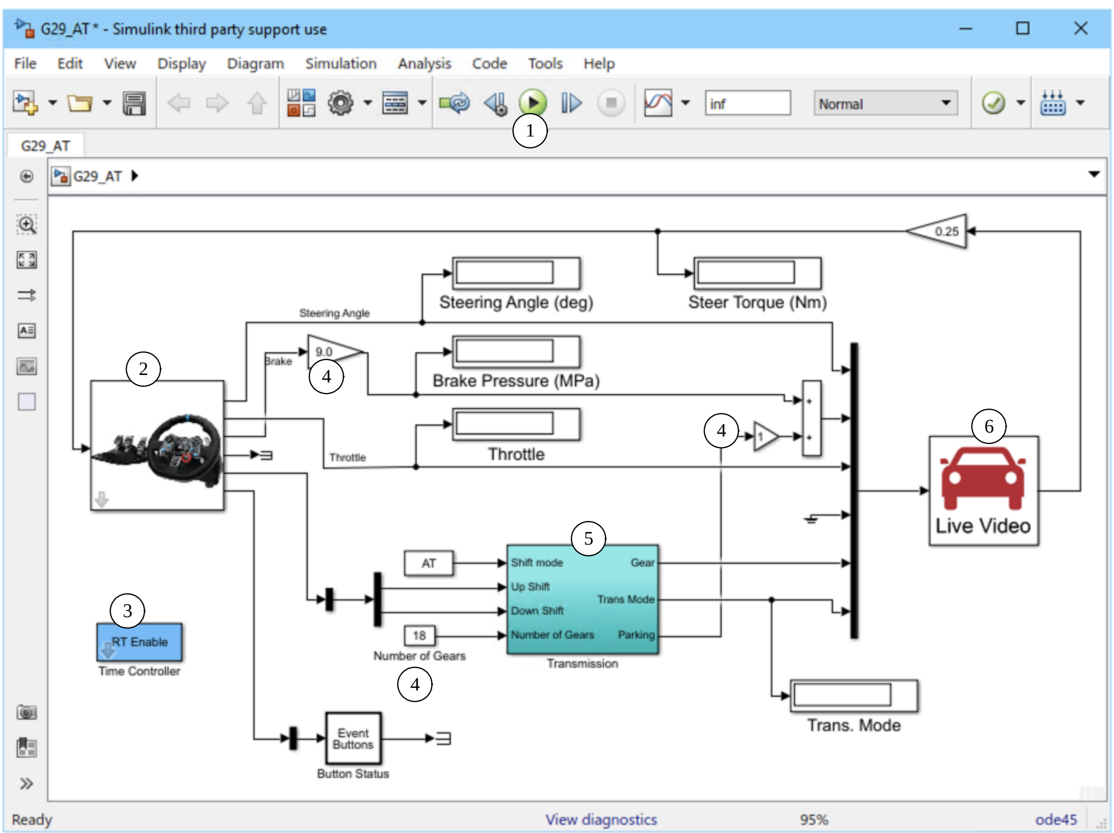


Figure 26. Simulink model for the G29 and a vehicle equipped with an automatic transmission.

(5) Transmission model. This model accepts as inputs from the Logitech hardware the commands for upshift and downshift (between Reverse, Neutral, and Drive), the maximum number of gears, and the shift mode (automatic or manual), and translates them into the transmission gear position, the transmission mode, and a parking control. The VS Math Model does not

have a built-in transmission Park mode, so the parking control represented here is an additional input via the brake pressure imported into the VS Math Model. The imported brake pressure is small and is intended to be sufficient to prevent the vehicle from moving without applying the Logitech brake pedal.

6 VS S-Function, representing either CarSim or TruckSim in the Windows DS simulation. Figure 27 shows the location of the VS S-Function in the Simulink Library Browser: The S-Function category is **DS S-Function** 1. With CarSim Windows DS the S-Function to use is **CarSim S-Function_video** 2. TruckSim Windows DS has an equivalent S-Function block.

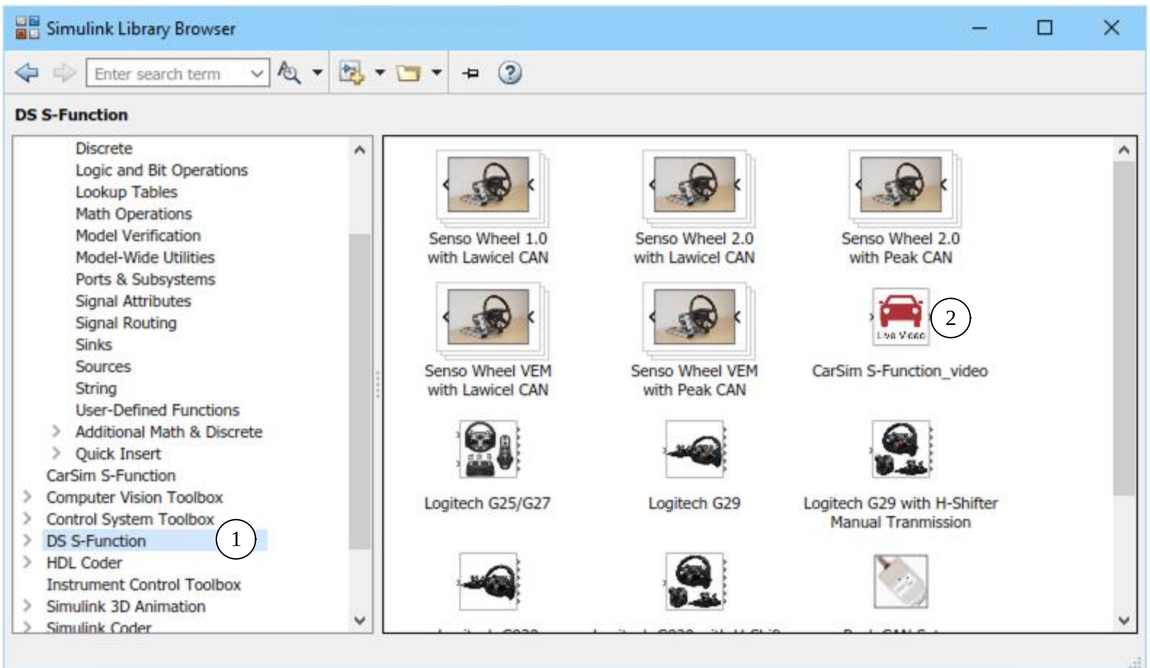


Figure 27. Simulink library of CarSim Windows DS S-Functions.

Support for HEV and EV Powertrains

Since the HEV (hybrid-electric) and EV (electric) powertrain options in CarSim and TruckSim do not have a geared transmission, running a Windows DS simulation with one of these vehicles will result in an error due to the import variable `Imp_Gear_Trans` not being recognized. Future releases of Windows DS will include both HEV and EV vehicles and this issue will be addressed via the shipping examples. For the 2021.1 release, a new dummy import variable can be created and placed on the Import Array in place of `Imp_Gear_Trans`, eliminating the error condition.

To do this:

1. Browse to the **I/O Channels: Import** dataset for your Windows DS example.
2. In the Misc. yellow field, type: `define_import imp_dummy` (Figure 28, 1).
3. At the top of the screen, use the drop-down menu to select the Windows DS **Run Control** dataset associated with this **I/O Channels: Import** dataset (2). This will generate a new

Import list containing the new import variable (for more information about using the options on the **I/O Channels: Import** screen, click the **Help** icon in the upper right corner of the screen).

4. From the list of **Available Variables**, locate the new dummy import variable (3) and add it to the Import Array in place of `Imp_Gear_Trans` (4). Be sure to remove `Imp_Gear_Trans` from the list.

5. Other Windows DS driver control settings may also need to be adjusted (e.g., brake gain).

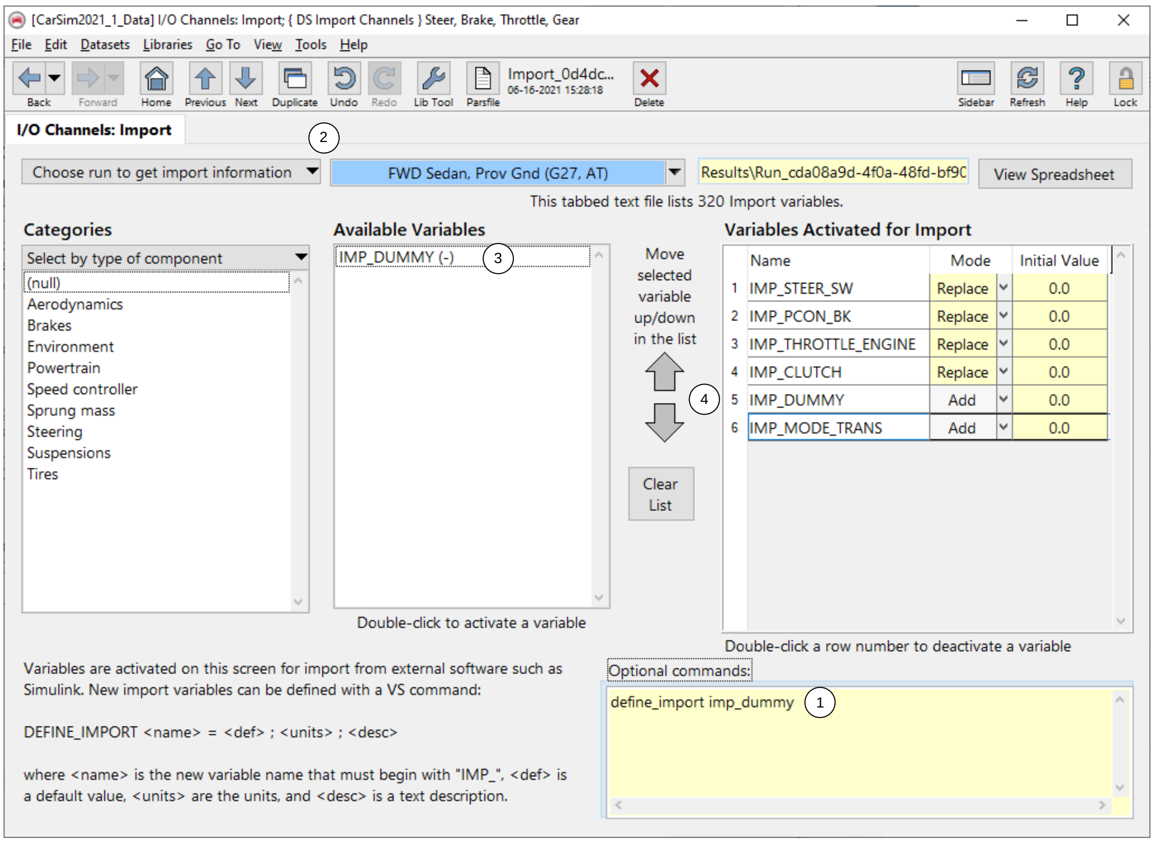


Figure 28. Windows DS Import Array with Dummy Import when used with HEV / EV vehicles.

Updates

This section includes updates as they relate to Windows DS. For more detail, see the Backward Compatibility document, available in **Help** > **Release Notes**.

> **Note** The Simple VS/STI and Simple TYDEX/STI tire models are not supported for use with Windows DS.

2021.1 vs 2021.0

In all prior versions, CarSim and TruckSim DS required separate downloads from the carsim.com website. Starting with version 2021.1, the DS tools and datasets are included in the installed

CarSim_Prog or TruckSim_Prog folders as CPAR archives that are used to build databases, as described in the first part of this document. This document is now included in the **Help** menu of the VS Browser.

All Windows DS examples have been updated to use the numerical integration option AB-2 with a math model time step of 2000 Hz. These settings, made on the respective **Models** screens, are now consistent with other shipping examples in the software that link to external models. This change necessitated minor updates to the Events used with the Proving Grounds; in particular, with respect to specifying some additional state variables for the vehicle and wheel speeds to ensure the vehicle remains in motion following a restart.

2021.0 vs 2020.1

CarSim and TruckSim have been receiving small adjustments to the animator asset scaling for various vehicle parts, including the 2A Conventional Van's cab used in TruckSim Windows DS. When databases built in 2020.1 are opened with 2021.0, these animator assets will initially have incorrect vertical offsets. The solution is to replace these datasets and associated shape files with what is shipped in 2021.0.

2020.1 vs 2020.0

The animator sound files for the engine and tires were moved to a new location in the CarSim_Prog and TruckSim_Prog folders: Resources\Animator\Audio_All_VS. For the 2020.1 release, new datasets have been created to refer to these new locations. For end-users who are upgrading from previous databases, please refer to the 2020.1 Windows DS databases for the new organization.

2020.0 vs 2019.1

Significant changes were made in the DS database for the 2020.0 release.

Animation shapes

Both CarSim and TruckSim have benefited from a multi-year effort focused on updating the graphical animation assets used throughout the product and rendered with VS Visualizer. As new animation shape files were created, previous generation animation shape files were moved to a dedicated folder within the CarSim_Prog and TruckSim_Prog folders to facilitate their eventual removal from the product altogether.

For the 2020.0 release, the datasets in CarSim and TruckSim Windows DS have been updated to reference the new animation shape files. For end-users upgrading their databases from previous versions, please take note of the new datasets in the 2020.0 release; migration efforts from the deprecated assets to their replacements should commence with this release of Windows DS, using the 2020.0 Windows DS databases as a reference.

Proving Grounds: Roads, Road Switching, Events, and RESTORE_STATE

For many years, the Windows DS Proving Grounds included a detailed set of Events that monitored the vehicle position and triggered a switch to a new road by comparing the global vehicle position relative to the current road and a potential new road. The new roads were also linked in the Events. This worked well but could result in occasional S-L errors (Station and Lateral Position), corresponding to multiple solutions for a given position (see **Help > Paths, Road Surfaces, and Scenes > Paths and Road Surfaces** for more information).

For the 2020.0 release of Windows DS, the road switching method on the Proving Grounds was updated to make use of Road Boundaries, a math model and GUI feature that was introduced in version 2016.0. The Road Boundaries greatly reduced the total number of Events and serve as a more general solution to vehicle position tracking, road switching, and resetting.

> **Note** The Event-based method used in previous versions still works if those databases are opened in new versions of the software. Please take note of any user-defined road and path IDs: they must be 999 or larger.

The new method works as follows:

1. At the start of the simulation, the VS Command `SAVE_STATE` is used to save the values of the state variables of the vehicle model. These are used later in case the vehicle goes out-of-bounds.

2. The first Event series (Figure 29 and Figure 30) is part of the Infinite Road: when the vehicle reaches the end of the road, its position is reset to the beginning (e.g. the vehicle's Sprung Mass Coordinate System Origin global position Xo = 1945 m) while the vehicle remains in motion. This Event series is only in effect when the vehicle is on the Infinite Road, indicated by crossing the boundary from the VDA South road to the Infinite Road.

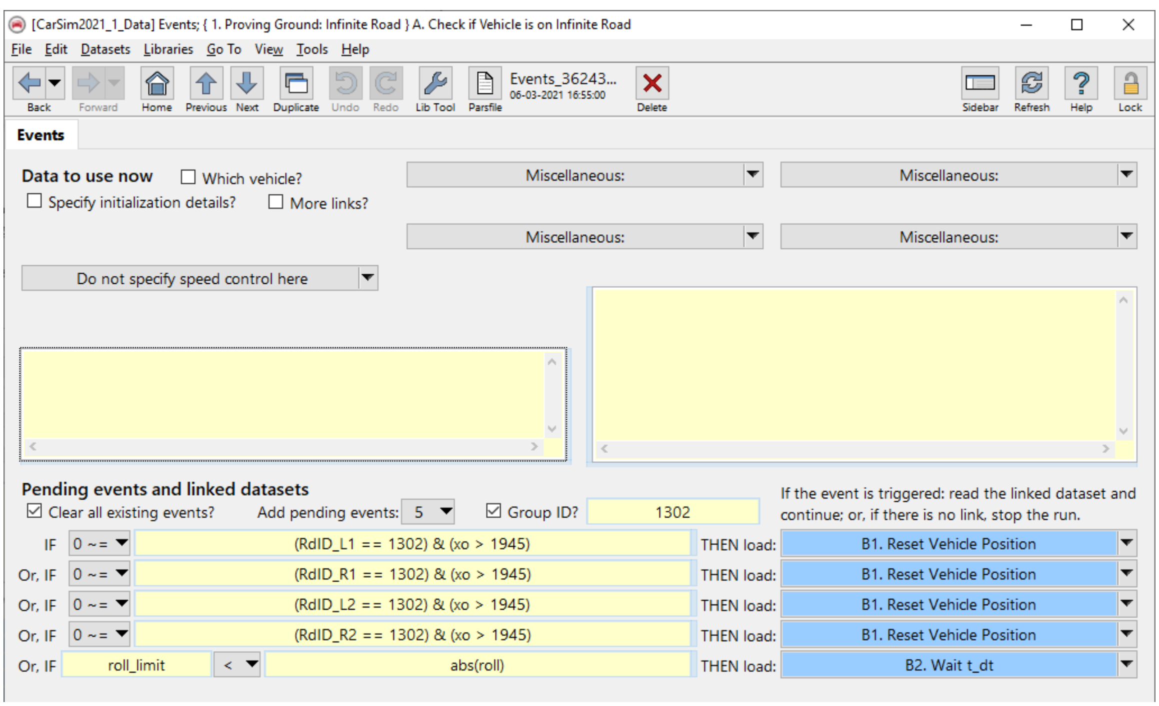


Figure 29. Event used to check vehicle position on Infinite Road.

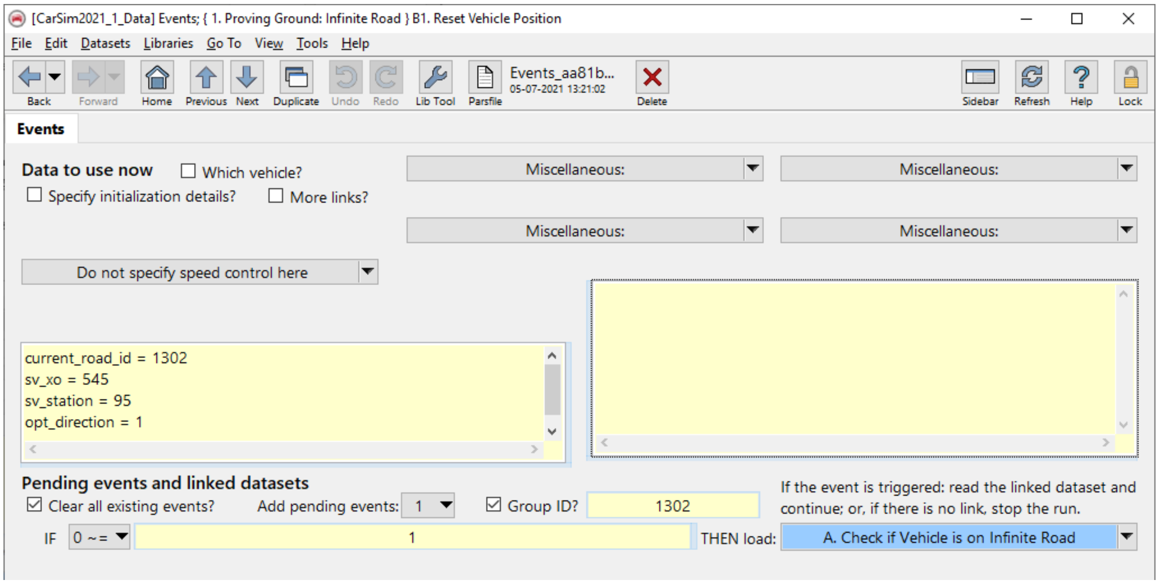


Figure 30. Event used to reset vehicle to the start of the Infinite Road.

3. The second Event series is always active and monitors the road IDs of the tires (Figure 31). When any of the tires cross a boundary, a switch occurs per the Road Boundary definitions. If the boundary leads to the intermediate road designated with ID 1234, this counts as an out-of-bounds condition and the vehicle is reset back to the center of the VDA South road, the same location where the vehicle was at the start of the simulation.

   The vehicle is considered out-of-bounds when it has traveled a set distance from the left or right-sides from most of the roads and is intended to catch cases of vehicle spinouts, roll-overs, and cases where the vehicle overshoots a road, e.g., driving too fast on the banked oval and it slides off the road. Since the Road Boundaries are set up to prevent the vehicle from traveling too far from the road, the occurrence of S-L errors (i.e., multiple solutions for Station-Lateral position) have essentially been eliminated.

The Event-based reset triggered by the tire road IDs uses a combination of the VS Command `RESTORE_STATE` and the setting of values of state variables in a Misc. yellow field in the Events. Although the `RESTORE_STATE` command will reset the state variables to their values when the simulation started, the order of operations in the simulation time step timeline is such that the restore happens at the end of the time-step. At the beginning of the next time-step, the saved states for variables set via the import array are discarded and subsequently assigned new values via the imported data. Since the vehicle is being controlled with the Logitech hardware and receiving data for throttle, brake, steering, and shifting via the import array, certain state variables must be explicitly set in the Event's Misc. yellow field to ensure a smooth restart.

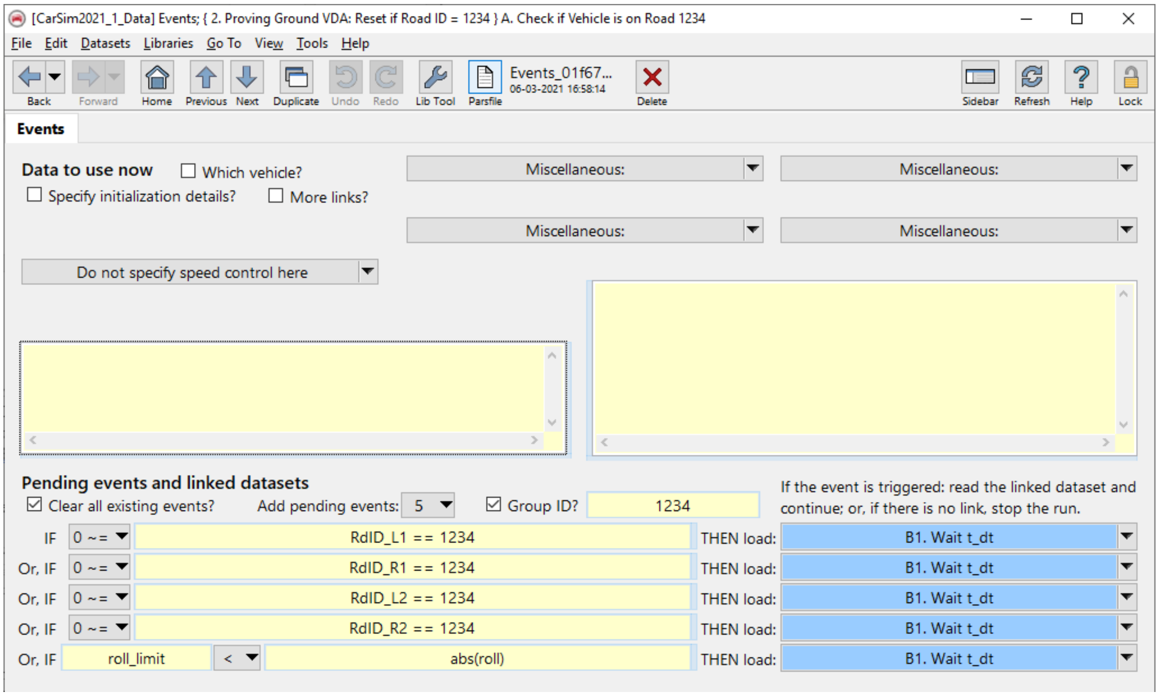


Figure 31. Event used to check tire position relative to Intermediate Road (ID = 1234).

The Road Boundaries are organized as follows, based on new roads that are overlaid relative to the existing Proving Ground roads:

High Speed Oval and VDA (Vehicle Dynamics Area)

- ID 1306: VDA North
- ID 1307: VDA South
- ID 1308: Oval East
- ID 1309: Oval West

Infinite Road

- ID 1302

The Handling Course is accessible from the VDA North road via a short link road. The IDs are:

- ID 1201: Handling Course
- ID 1303: Link Road

Ride Road

- ID 1305

Intermediate and Transition Roads

- ID 1234: Out-of-bounds; resets vehicle to VDA South (ID 1307)
- ID 2234: Intermediate Road between the entry and exit of the Ride Road (ID 1305)

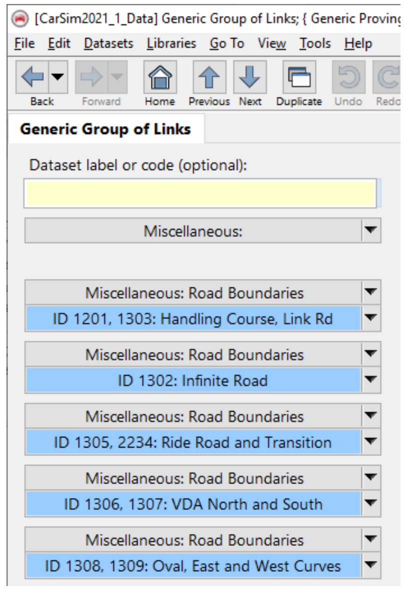


Figure 32. Group of Road Boundaries used for the Windows DS Proving Ground.

Since all roads are defined using a set of global X-Y coordinates for their reference line, the location where two roads meet are known as they will share a set of (or have very closely located) global X-Y coordinates, while the Station values for each road's set of X-Y coordinates are calculated by the VS Math Model. This lack of ambiguity makes for clean Road Boundary transitions between the VDA North and the Link Road, and the VDA South and the Infinite Road.

For example:

- When the vehicle is at a location on the VDA North road corresponding to the start of the Link Road to the Handling Course, the boundaries allow the vehicle to switch to the Link Road. The same is true when the vehicle is on the Link Road and it drives back onto VDA North.
- Similarly, when the vehicle is at a location on the VDA South road corresponding to the start of the Infinite Road, the boundaries allow the vehicle to switch to the Infinite Road. The same is true when the vehicle is on the Infinite Road and drives back onto VDA South.
- If the vehicle crosses the boundaries at one of these shared locations (X, Y), the initial Station on the branch roads (i.e., Link Road or Infinite Road) is 0 m, relative to the Station position on the old road (VDA North or VDA South, respectively).
- If the vehicle tries to cross the outer lateral boundaries for either VDA North or VDA South and the vehicle is not within the transition range to either the Link Road or the Infinite Road, the vehicle will experience an out-of-bounds condition, trigger the switch to the transition road (ID 1234), and then be reset to the center of VDA South.

Setting up the Road Boundaries to transition from VDA South to the Ride Road is more challenging: as with the Link Road and the Infinite Road, the Ride Road is not looped and therefore has a defined starting position and ending position (global X-Y coordinates with corresponding Station). However, unlike these other two roads, the Ride Road's beginning and end coordinates both link to VDA South and are also very close together. This results in their right-hand side boundaries being very close to one another.

If the only requirement was to enter the Ride Road directly from VDA South, the Road Boundary definitions would not be difficult: VDA South has two sets of valid ranges of Station for each of the Ride Road's entry and exit: if the vehicle is within one of these ranges and is in a lateral position corresponding to the Ride Road's entry or exit, a road switch would occur and therefore no ambiguity would exist.

The Ride Road also has Lateral Boundaries (one set of boundaries on each side, each as a function of Station) to prevent the vehicle from going too far off course, and this is where the complication arises: we want to maintain the Ride Road's lateral boundaries, but we do not want to switch to the intermediate road (ID 1234) and reset to the center of VDA South if the vehicle is driven directly between the entry and exit sections of the Ride Road (i.e., rather than first driving onto VDA South as an intermediate step).

When the vehicle is driven directly between the Ride Road's entry and exit (crossing the right-hand side lateral boundary), we have not exited the road — i.e., it is still ID 1305 — but the road's Station positions are vastly different. We cannot use the Road Boundaries to directly switch from one position on a given road to another position on the same road —i.e., we cannot use a boundary for ID 1305 to switch to another position on ID 1305 — and we do not want to switch to the intermediate road (ID 1234). Therefore, another solution is needed.

To avoid the ambiguity:

- An intermediate road (Figure 33, ID 2234) is defined to be between (and is parallel to) the entry and exit of the Ride Road. It is also adjacent and perpendicular to VDA South.
- If the vehicle drives directly between the Ride Road's entry and exit, it must first cross ID 2234. Since road ID 2234 has its own left and right boundaries (Figure 34 and Figure 35) and these are parallel to the right boundaries of the Ride Road (albeit at different Station positions), distinct boundary conditions exist for each of these two cases. Therefore, they can be managed as two unique sets of boundaries, with two unique sets of relative Station positions.
- Since the Intermediate Road (ID 2234) can be directly accessed from VDA South, the vehicle can also enter the Ride Road by first driving onto this Intermediate Road.

This method helps to ensure a positive transition while also properly managing the new Station position when the vehicle transitions between the entry and exit of the Ride Road. Figure 34 and Figure 35 show a few of the road boundary datasets used as part of the Proving Ground.

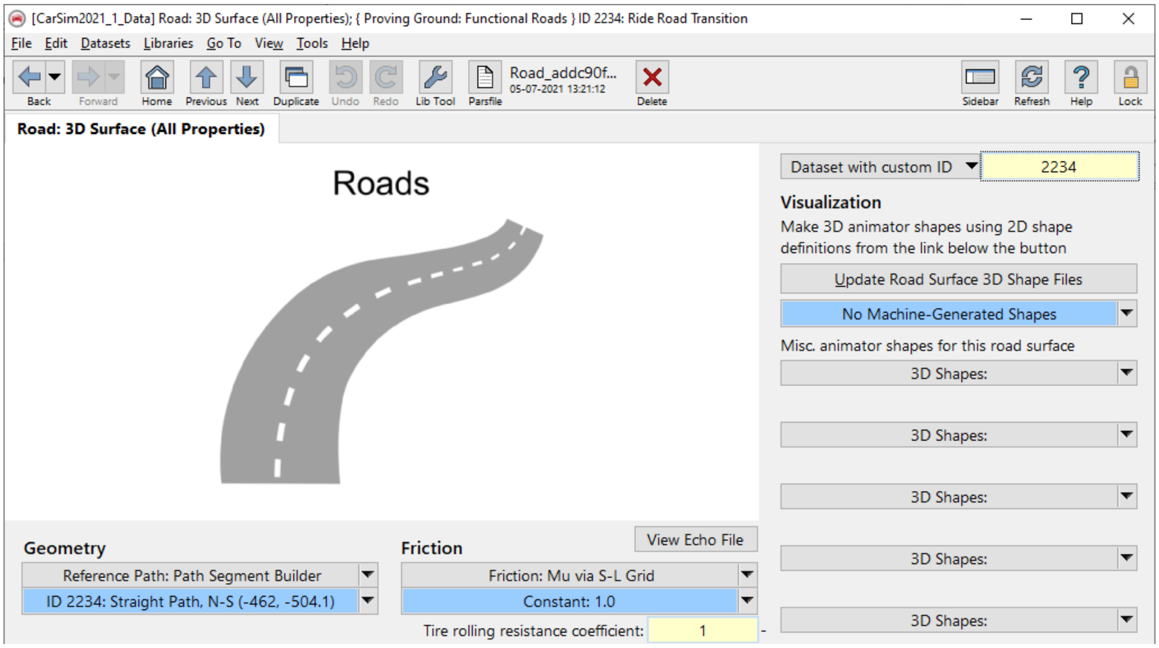


Figure 33. Ride Road transition road (ID = 2234).

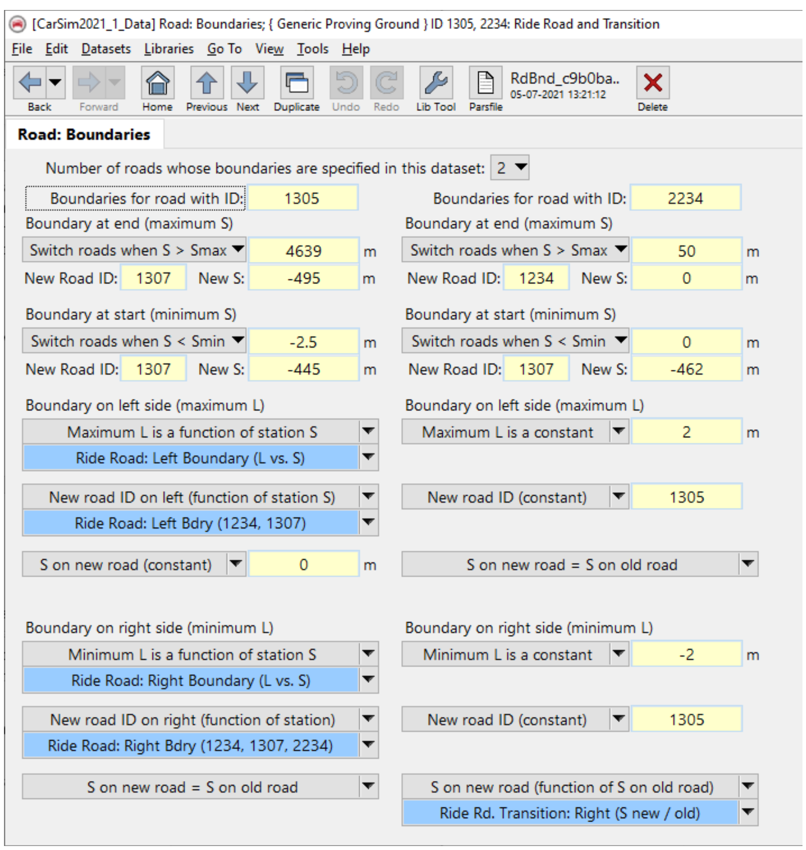
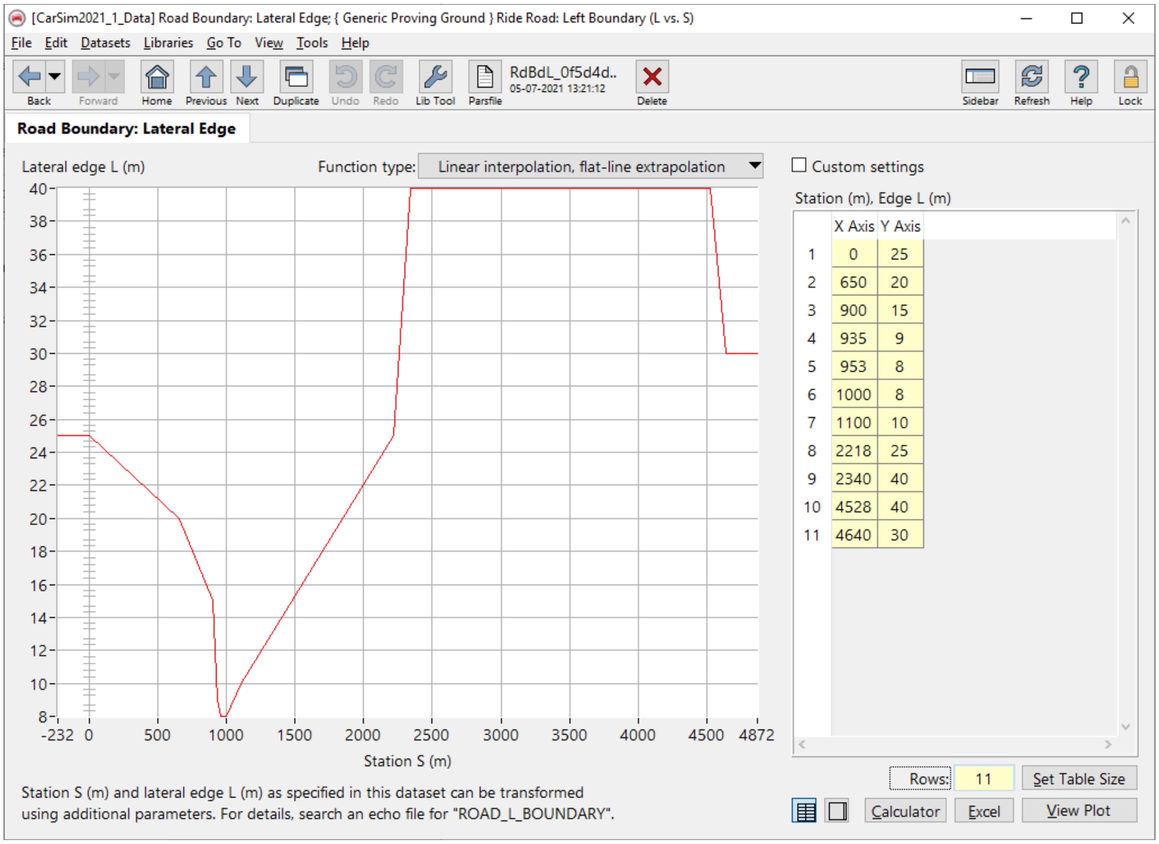


Figure 34. Road Boundary dataset for the Ride Road (ID = 1305) and Transition (ID = 2234).



	X Axis	Y Axis
1	0	25
2	650	20
3	900	15
4	935	9
5	953	8
6	1000	8
7	1100	10
8	2218	25
9	2340	40
10	4528	40
11	4640	30

Figure 35. Lateral Edge vs. Station boundary for the left side of the Ride Road.

Animator: Camera Setup

The Window DS examples typically use an in-vehicle view position whose data is set using the **Animator: Camera Setup** screen (i.e., Vehicle occupant point of view), linked in the upper right corner of the **Run Control** screen (Figure 12, 6). This makes sense for a Driving Simulator environment since the intent is to represent the view position from the perspective of the driver.

For the 2020.0 release of Windows DS, a new **Animator: Camera Setup** dataset is included (one each in CarSim and TruckSim) whose position is placed at the left rear quarter of the vehicle. This viewing position, to the left of and behind the vehicle, is especially useful as a diagnostic tool when setting up and evaluating the Road Boundaries used for the road switching on the Proving Grounds.

Since the Windows DS examples with the Logitech hardware are intended primarily for use as a desktop Driving Simulator rather than on a motion platform such as a hexapod, the end-user does not have the same sense of vehicle motion, i.e., the vehicle motion can be seen, but not felt. Therefore, an exterior view of the vehicle can serve as a partial substitute for the lack of a motion base platform by providing the end-user with additional visual cues regarding the vehicle in relation to the road.

VS Visualizer HUD (head-up display) with Road ID Data

As noted, the 2020.0 release of Windows DS uses a combination of Road Boundaries and Events to monitor which road the vehicle is on and either switch to an adjacent road, or catch the vehicle leaving the road and return it to VDA South in a restored state.

To help the end-user track which road the vehicle is on and monitor when a particular road switch has occurred, a **Generic Data Lists** dataset includes output variables that are visible during the run in VS Visualizer as HUD elements. Although this dataset is not required for use in the simulation, the information displayed is very useful when learning how the Road Boundaries work.

For this release, this dataset is linked to the **Run Control** screens for the simulations using the automatic transmission-equipped vehicles on the Proving Ground. It can, however, be linked to any run that includes Road Boundaries.

Vehicles

The D-Class Sedan in CarSim Windows DS has been updated to match the equivalent vehicle included in the core CarSim database. This includes new high-resolution animation shapes for the exterior and interior.

The Heavy Duty Utility Truck and the City / Off-Road Course in TruckSim Windows DS have been removed to allow the product to focus on the Proving Ground and Mcity environments. As with any database, however, a CPAR file containing the Heavy Duty Utility Truck and/or the City / Off-Road Course may be imported into the TruckSim 2020.0 Windows DS database after exporting them from either the core TruckSim database or a previous release of TruckSim Windows DS.

> **Note** As with any driving simulator simulation, the expectation should be that vehicle data may need to be adjusted relative to its off-line simulation counterpart based on the desired driving simulator experience.

2019.1 vs 2019.0

The following changes were made in the vehicle data for this release.

1. For CarSim Windows DS, the FWD and RWD sedans previously exhibited lift-throttle oversteer and on-throttle oversteer, respectively. To improve the vehicle's response to these throttle inputs, the longitudinal-force toe and lateral-force steer compliance coefficients in both the front and rear compliance datasets were updated.

2. For TruckSim Windows DS, the 2-axle Conventional Van includes redesigned shift schedules to reduce gear-hunting and some erratic shift behavior.

3. The mass of the wheel/tire assembly may now be entered on a tire screen, rather than being included with the suspension’s unsprung mass. This means various tires may be used with a single suspension kinematics dataset without having to recompute and reenter the unsprung mass values. Consequently, every suspension kinematics dataset, independent or solid axle, has been adjusted to account for the removal of tire mass, and every tire dataset

has been assigned a mass value. Wheel/tire assembly spin inertias were also assessed and updated as part of this process.

Backward Compatibility

CarSim and TruckSim Windows DS versions 2019.1 and newer are backward compatible to version 2017.1.

This section describes the steps necessary to run a Windows DS database in CarSim and TruckSim 2019.1 or newer that was originally built in either 2018.1 or 2017.1. A detailed set of Backwards Compatibility information is available in a separate Technical Memo (**Help > Release Notes**).

Animator: Sound Modifier Dataset has Unbalanced Parenthesis

One of the **Animator: Sound Modifier** datasets included with Windows DS is used to represent the sound a tire makes when it is subjected to a lateral slip angle. A VS Command equation is used to calculate an output variable **mod2** (Figure 36, 4), which in turn is used by VSV.

As noted in Figure 36, there is a missing closing parenthesis (5). Based on the version of CarSim or TruckSim being used, the method of handling this is different.

1. In CarSim and TruckSim 2018.1 and earlier, the VS Math Model would read this equation and automatically add the closing parenthesis.
2. The release of 2019.0 includes more aggressive error handling intended to catch such instances in VS Command expressions, with one side effect being that the VS Math Model will no longer automatically add the closing parenthesis to this equation. When a simulation is executed and unbalanced parenthesis exist in VS Command expressions, an error dialog appears, and the simulation will stop. The VS Commands will have to be corrected.
3. For CarSim and TruckSim 2019.1, the **Tire** screen no longer includes a dedicated link for the **Animator: Sound Set** datasets. For end-users wishing to use the tire sounds, the recommendation is to use the checkbox **Custom settings** in the center of the **Tire** screen. In addition to a Misc. yellow field, a library link becomes available and **Animator: Sound Set** datasets can be linked.

From an upgrade standpoint, the end-user has two choices: add the closing parenthesis either before or after opening the 2018.1 Windows DS database with 2019.0. Correcting the VS Command equation before the database conversion is preferred.

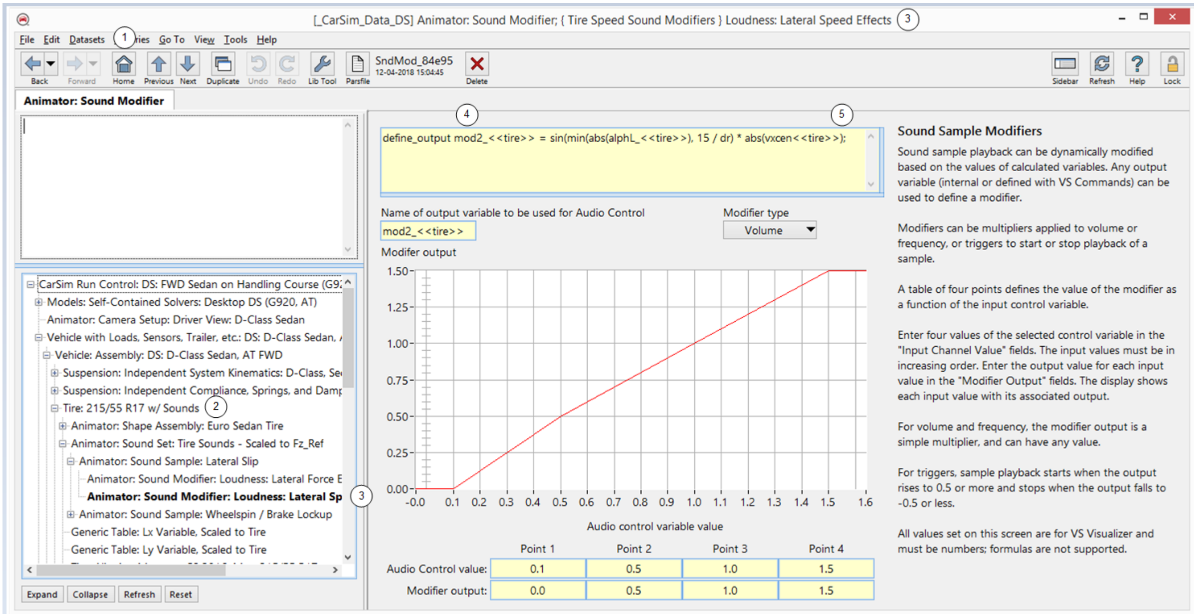


Figure 36. Animator: Sound Modifier dataset for Loudness: Lateral Speed Effects.

1. **Home** button.
2. Tire dataset to which this **Animator: Sound Modifier** dataset is linked.
3. **Animator: Sound Modifier** dataset for the tire's **Loudness: Lateral Speed Effects**.
4. Output variable name `mod2` and corresponding VS Command equation used to calculate the tire sound based on lateral slip angle.
5. Location of the unbalanced parenthesis. A closing parenthesis is missing at the end of the equation; it should be …abs(vxcen<<tire>>)), as in:

```
define_output mod2_<<tire>> = sin(min(abs(alphL_<<tire>>), 15
/ dr) * abs(vxcen<<tire>>));
```

To correct the issue, use the option **Find Text in the Database …** (Figure 38) to locate the output variable `mod2` (Figure 37), then browse directly to the **Animator: Sound Modifier** dataset.

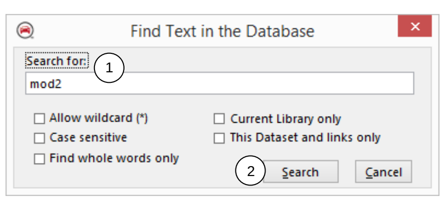


Figure 37. Find Text in the Database window, with search criteria set to mod2.

Type `mod2` in the search field (1), the click the **Search** button (2).

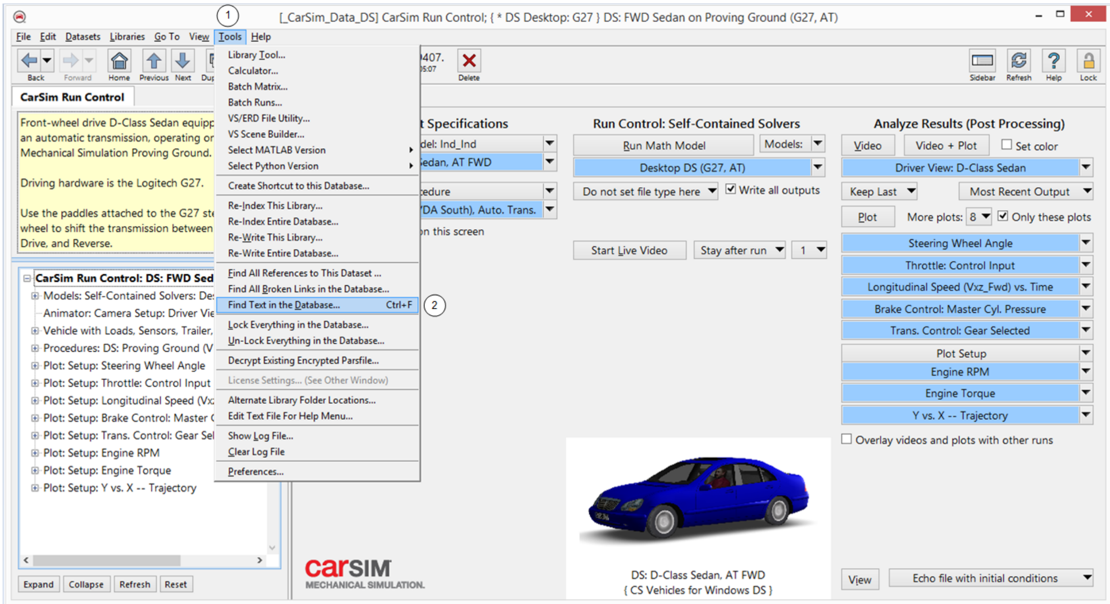


Figure 38. Run Control screen appearance of Windows DS, showing the Find Text option.

The Windows DS database contains two references to the output variable **mod2** (Figure 39, 1). Double-click on the second instance, located in **Tire Speed Sound Modifiers**. The button **Find References to the Current Dataset** will become active (2). Click this button to browse to the selected dataset (Figure 40).

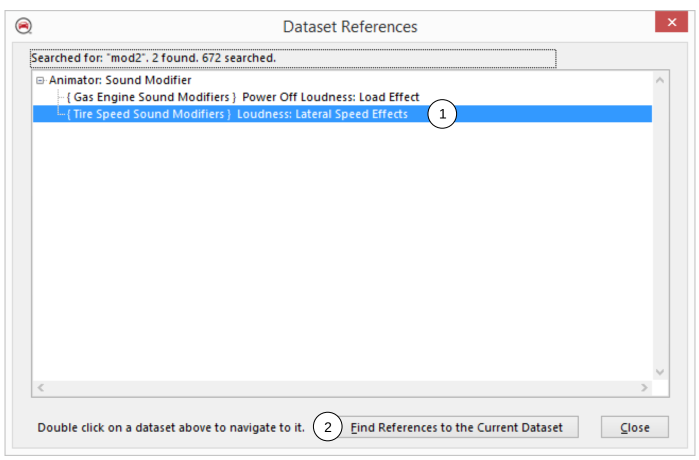


Figure 39. Dataset References to mod2.

As noted above, modify the VS Command equation by adding a closing parenthesis at the end of the expression, then click the **Home** button (Figure 40, 1) to return to the **Run Control** screen. Since the same tire dataset is used for all four tires, we need only modify this **Animator: Sound Modifier** dataset once.

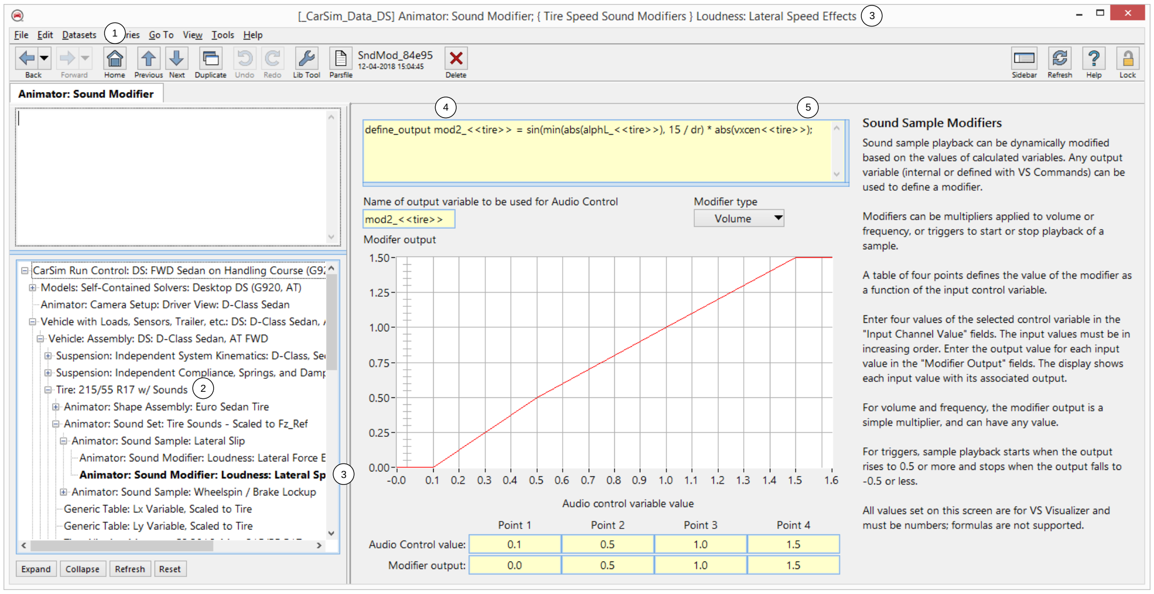


Figure 40. Animator: Sound Modifier dataset for Loudness: Lateral Speed Effects.

Invalid Settings for Fixed Time Step with Simulink

When running a Windows DS example with Simulink, an error may occur in which Simulink indicates that there is an invalid setting for the fixed-step size, and that all sample times in the model must be an integer multiple of the fixed-step size (Figure 41).

To correct this, browse to the **Models: Simulink** dataset linked from the **Run Control** screen and make sure the settings are as shown earlier in Figure 25 (page 19).

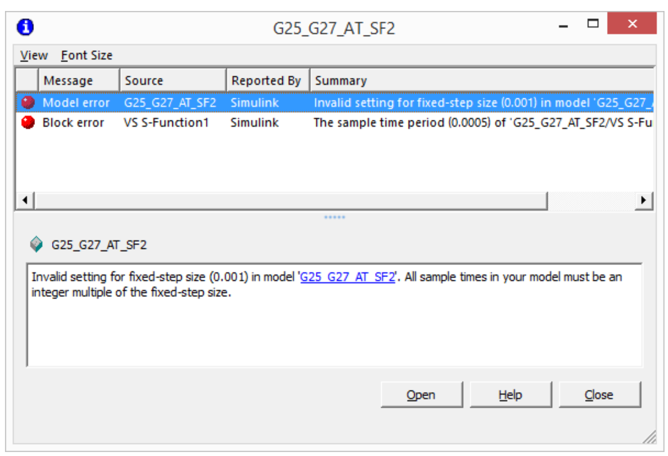


Figure 41. Simulink error dialog regarding a mismatch in time step sizes.

VS S-Functions

The VS S-Function has been updated for the 2019.0 release of CarSim and TruckSim. When performing a database conversion from 2018.1 or 2017.1 to 2019.0, the Simulink models will have to be manually updated to use the new S-Function.

Launch the Simulink Library Browser and click DS S-Function (Figure 42, 1). If it appears with two VS S-Functions (2, 3), the Simulink Libraries will need to be updated.

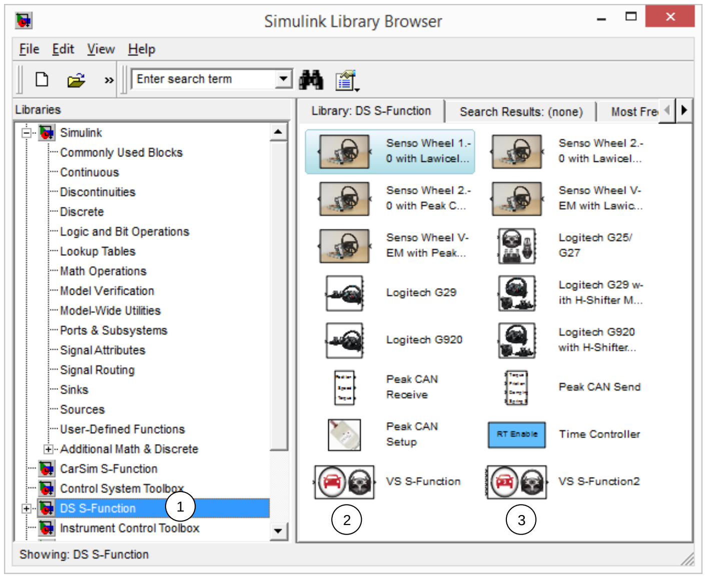


Figure 42. Simulink Library Browser showing Windows DS S-Functions, pre-2019.0.

This is a two-step process:

1. Locating the Simulink libraries in the 2019.0 Windows DS installation and copying them to the databases converted from 2018.1 or 2017.1, and
2. Deleting one of the pre-2019.0 S-Functions from the Simulink model and dragging in the new block.

Before beginning this process, close CarSim and MATLAB / Simulink.

1. From the installed 2019.0 Windows DS database, browse to the folder location `Extensions\Simulink_DS` (Figure 43, 1).
2. Two folders exist containing the Simulink libraries: `DS_SF_Library` (2) and `DS_SF_Library84` (3). Copy these two folders and their contents from the 2019.0 database and paste to the same folder locations in the databases converted from 2018.1 and/or 2017.1. Be sure to completely replace the previous folder content in the converted databases.
3. Launch 2019.0 Windows DS and open the database converted from 2018.1 or 2017.1.
4. Select one of the Windows DS **Run Control** examples that uses Simulink, and click the button **Send to Simulink**. When the Simulink model launches, it should appear as shown in Figure 44, with a Bad Link in place of the VS S-Function (1).

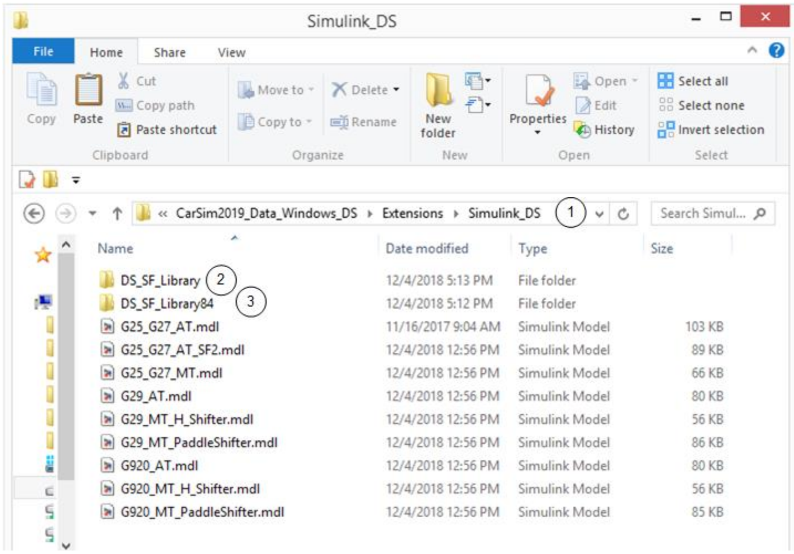


Figure 43. Simulink_DS folder, containing libraries and Simulink models.

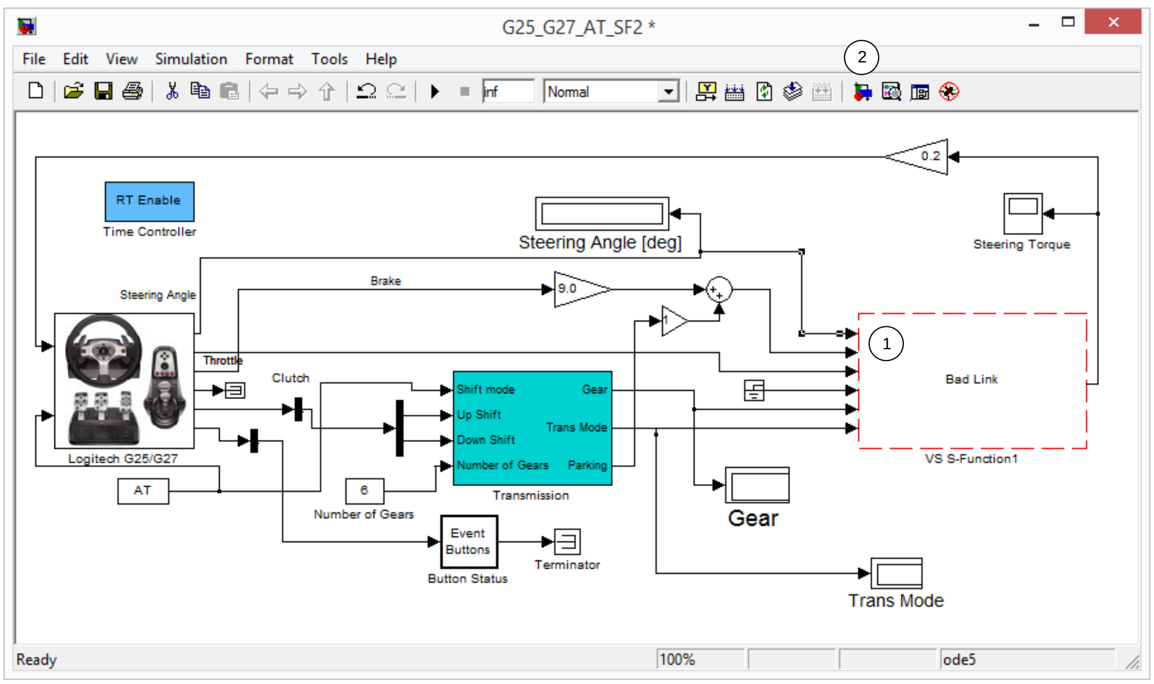


Figure 44. Simulink model from converted Windows DS database, after update of Simulink libraries.

Launch the **Simulink Library Browser** (Figure 44, 2) and click on the category DS S-Functions (Figure 45, 1). In place of the two VS S-Functions shown before (Figure 42, 2, 3), we should now see a single VS S-Function (Figure 45, 2).

1 DS S-Function library

2 VS S-Function

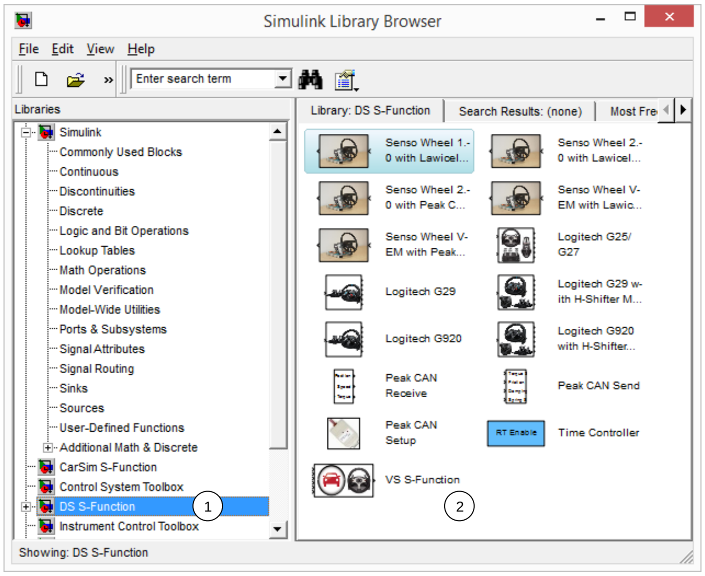


Figure 45. Simulink Library Browser, showing the VS S-Function built for 2019.0.

Delete the Bad Link (Figure 44, ①), drag the new VS S-Function into the Simulink model (Figure 46, ①), and reconnect the channels, being sure to get them in the correct order per the Import Array. After these changes, the Simulink model should appear much like Figure 46.

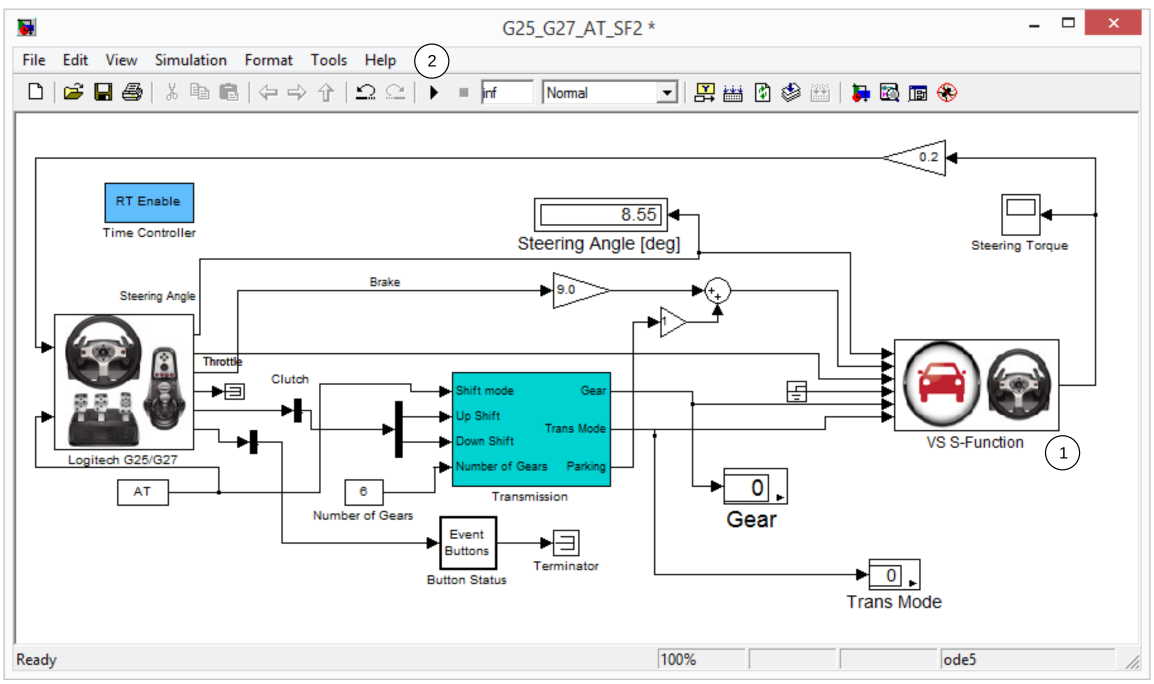


Figure 46. Simulink model after updated to use VS S-Function for 2019.0.

Click the Simulink **Play** button (Figure 46, ②); the simulation should now run correctly.

Troubleshooting

Yaw angle for lead unit differs greatly relative to yaw of trailer(s)

When running a combination vehicle in some scenarios where the vehicle follows a looped path, you might notice that the yaw angle of the vehicle lead unit can go outside the range of ±180°, whereas the yaw angle of any trailer is always limited to ±180°. This is because the tractor yaw is calculated by integrating the yaw rate. The integration allows the values to go outside the range of ±180°. On the other hand, trailer yaw is calculated using an arctan function that always returns a value with the range of ±180°.

Compatibility between drivers for Logitech G27 and G29 / G920

The Logitech G27 and G29 / G920 drivers have historically not compatible with each other. Although a set of examples are included for each of the three wheels, development exercises initially indicated that the end-user could only have one set of Logitech drivers installed at a time.

"Joystick Not Found" when starting a simulation with G29 / G920

When using the Logitech G29 or G920, the Logitech Gaming Software must be launched prior to using the wheel. If the end-user launches Windows DS, selects one of the G29 / G920-based examples, and tries to run the math model without having the Logitech Gaming Software running, an error dialog may appear indicating that the joystick is not found.

G920 "Logitech Gaming Software does not recognize drivers"

Although the Logitech drivers for the G29 and G920 are the same, the Logitech Gaming Software may not recognize the drivers when using the G920. If this occurs, please perform the following steps:

1. Close Windows DS along with any supporting software for the examples (i.e., VSV, MATLAB/Simulink)
2. Close the application Logitech Gaming Software
3. Unplug the Logitech G920's USB cable from your computer
4. Uninstall the Logitech drivers
5. Run the program `WheelDriverCleaner.exe` provided by Logitech.
6. Reinstall the Logitech drivers
7. Launch the Logitech Gaming Software program
8. Plug the Logitech USB cable into your computer
9. If the action of running `WheelDriverCleaner.exe` and reinstalling the Logitech drivers was successful, the G920 should be recognized by the Logitech Gaming software.
10. Relaunch Windows DS and select a G920-based example.

Note The Logitech drivers and the program `WheelDriverCleaner.exe` are included in the Windows DS databases, in the folder `Extensions\Windows_DS_Drivers`.

G920 PC exhibits blue screen behavior

If this behavior occurs, please run the `WheelDriverCleaner.exe`, included in the `Windows_DS_Drivers` folder (above).

G27 settings for degrees of rotation

If an end-user installs the G29 / G920 drivers, then uninstalls these drivers and installs the G27 drivers, it might be necessary to reset the **Degrees of Rotation** for the G27 to 900 degrees.

VS Visualizer does not launch when starting a Windows DS simulation

If VS Visualizer does not launch when a Windows DS simulation is started, check the following:

1. The number of live server connections is set to 1 on the **Run Control** screen (Figure 47, 1).
2. Open the **License Settings** window (**Tools** > **License Settings**) and make sure that the license option "Extra Live Animator" is checked (Figure 48, 1, 2).
3. On the **Models** screen, the keyword `LIVE_SERVER` is set to localhost (Figure 49, 1).

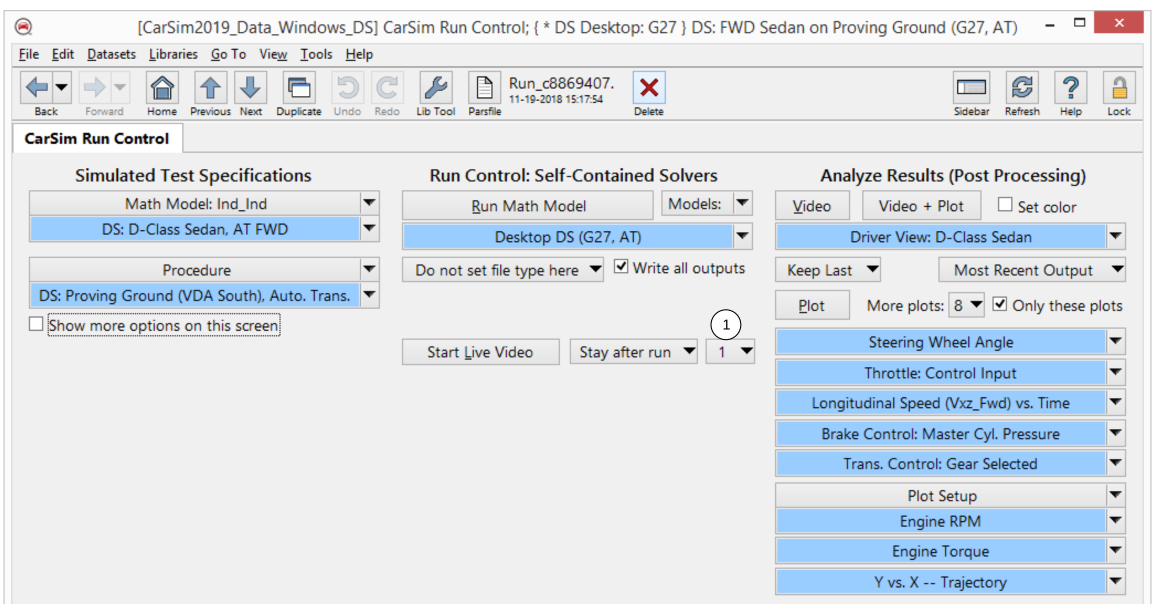


Figure 47. Run Control screen appearance with controls set up for Live Video data streaming.

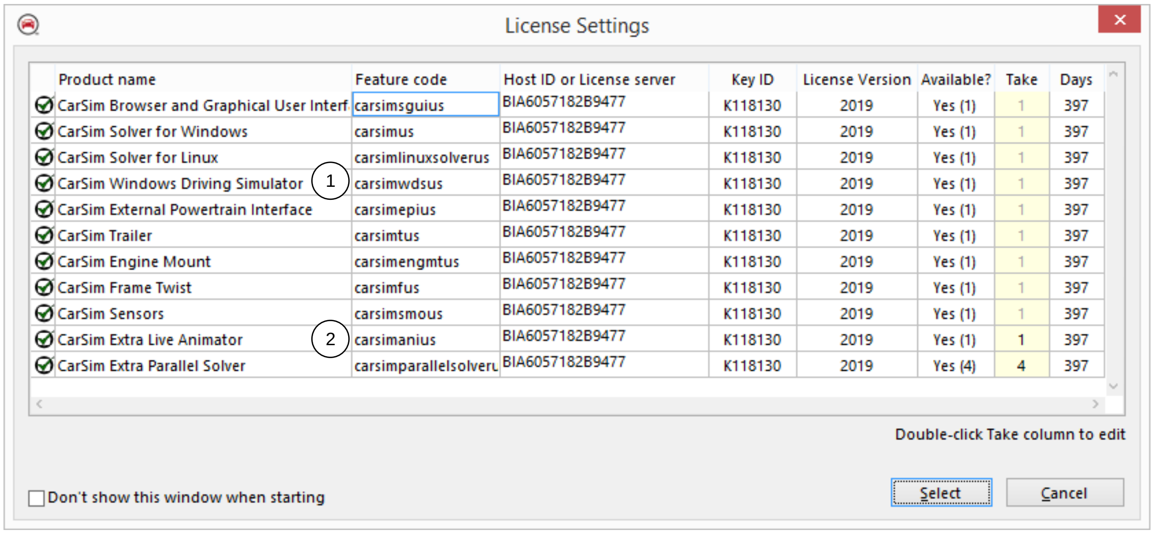


Figure 48. License Settings with License Features checked for Windows DS and Live Animator.

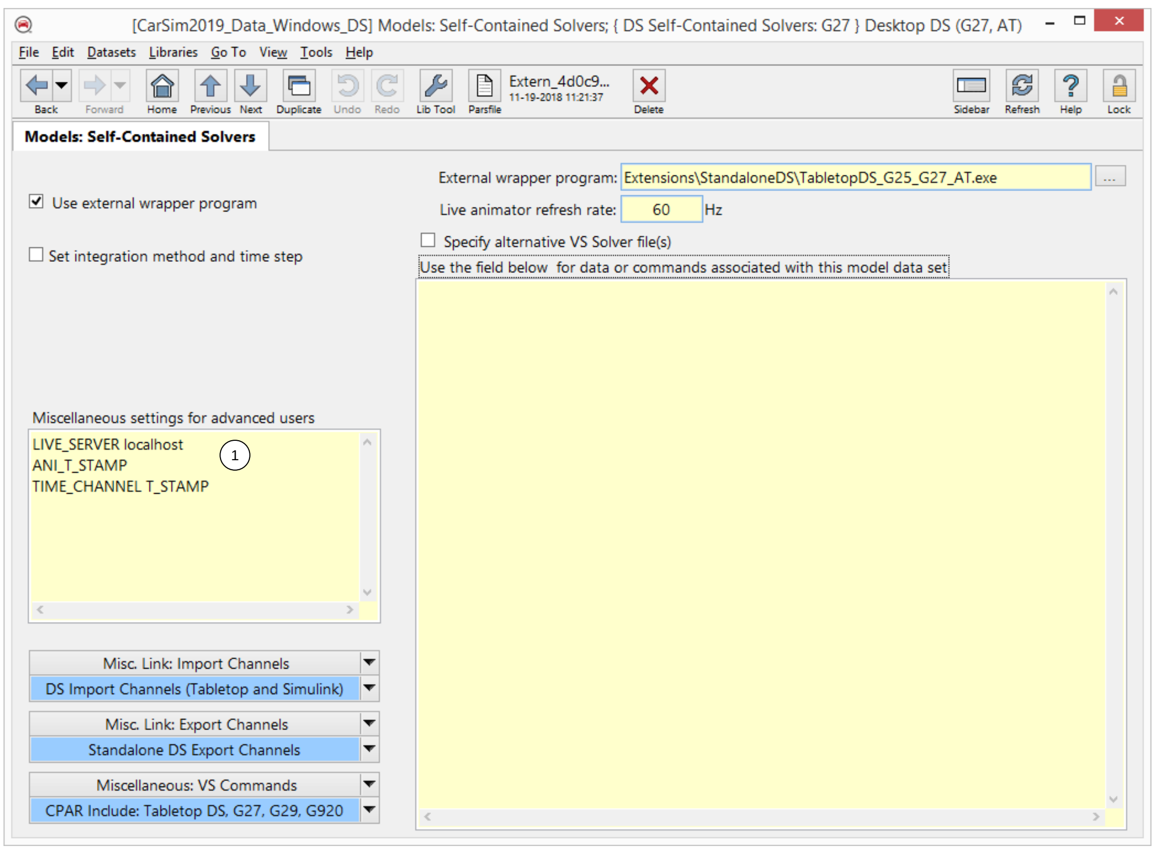


Figure 49. Models screen appearance, with VS Commands for Live Animation.