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

Convert a Simulink model to an Executable Application

This memo is a tutorial that demonstrates how to convert a Simulink model with a VehicleSim S-Function, `vs_sf`, to an executable application. You can run the file or distribute it to other users.

Simulink Coder (formerly RTW) generates C/C++ code from Simulink models, Stateflow, and MATLAB functions and compiles them into an executable application. The generated source code can be used for real-time and non-real-time applications. It can be run outside MATLAB and Simulink.

A simple antilock brake system (ABS) example is used in various forms in BikeSim, CarSim, and TruckSim. This memo is not intended for specialists in the Simulink model. Rather, the point is to show how to set up a VS vehicle model and the Simulink environment.

This tutorial requires Simulink/Simulink Coder (from The MathWorks), C compiler for Matlab/Simulink, and your installed VehicleSim product. The document assumes familiarity with Simulink/Simulink Coder. You should also already have a basic understanding of how to use the VehicleSim product with Simulink (refer to the **Help** document "Running a VS Math Model in Simulink").

Find the Existing Simulink ABS Example

The tutorial is based on an example Simulink ABS model that is used for several VehicleSim product database runs. The installed **Run Control** library has a category that includes Simulink runs (**Simulink and LabVIEW Models** in CarSim and TruckSim; **Simulink Models** in BikeSim). This category contains ABS examples: **Ext. ABS: Split Mu** (Figure 1) in CarSim and TruckSim; **ABS: Braking in Turn – Low Mu** (Figure 2) in BikeSim.

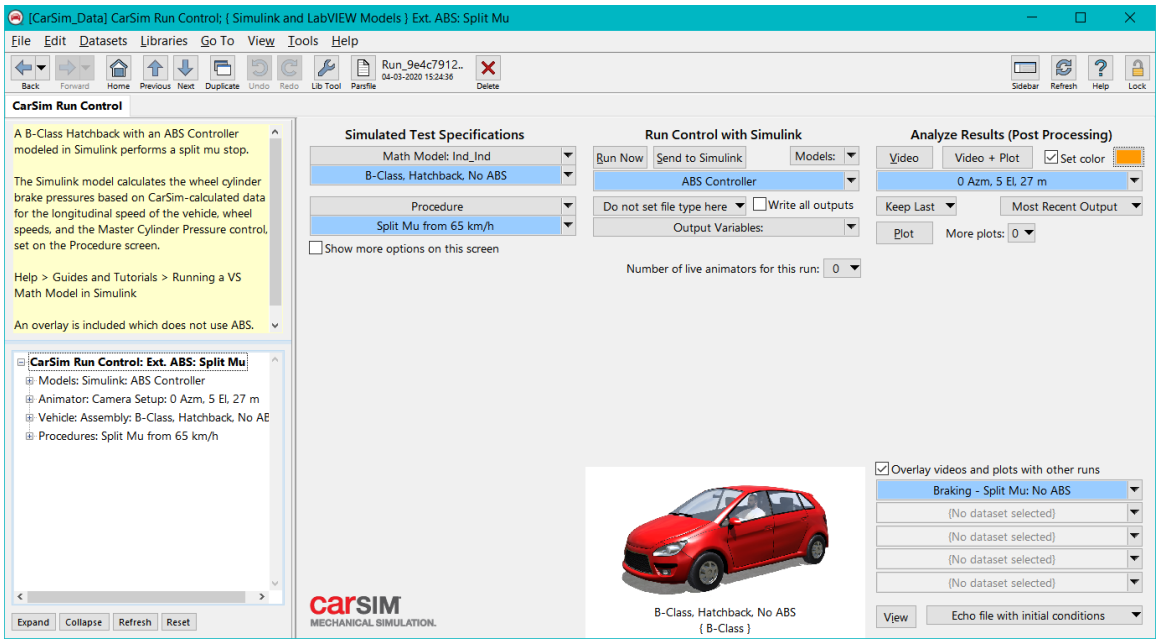

Figure 1. Run Control screen for ABS test on a split-mu surface.

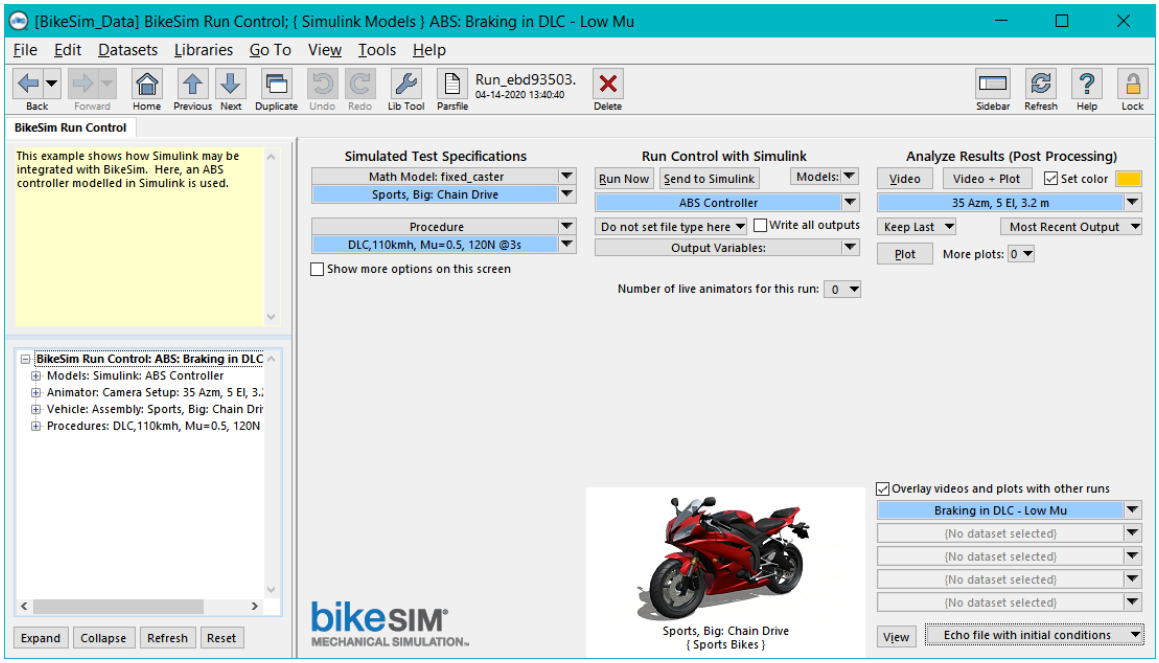

Figure 2. Run Control screen for BikeSim ABS test in a turn.

Copy S-Function Source File to Model Folder

The VehicleSim API and its examples can be found within the VS SDK. The VS SDK is a collection of tools and libraries intended to provide a development framework for interacting with the VehicleSim family of products. The VS SDK can be downloaded from the Mechanical Simulation website.

Copy the source file "`vs_sf.c`" from VS SDK to the folder of the Simulink model.

Setup Simulink Model Parameters

1. Check **Identify Simulink working directory** and Simulink folder (1) (Figure 3).
2. Setup the integration method and time step (2).

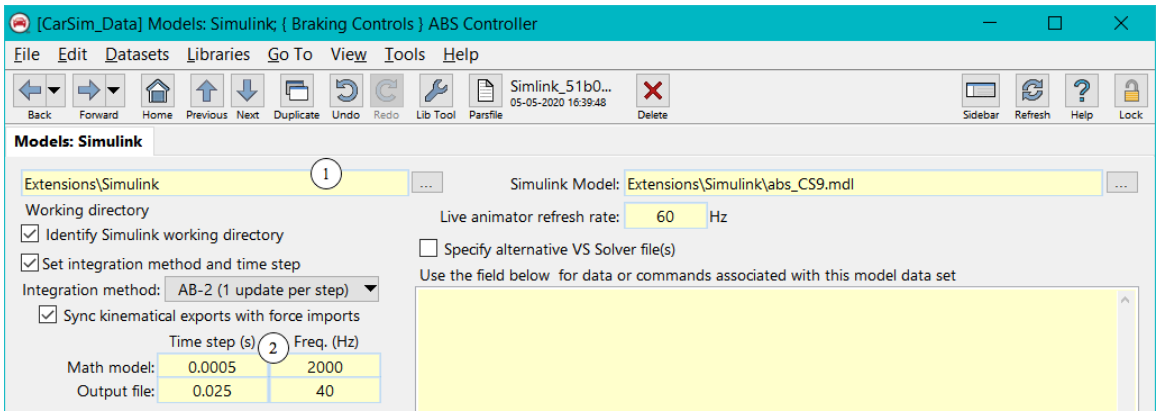

Figure 3. Setup Simulink Model Screen.

Setup and Build the Simulink Model

1. Click the **Send to Simulink** button (1) (Figure 4). The VS Browser will load the Simulink model (Figure 5), using updated connection information.

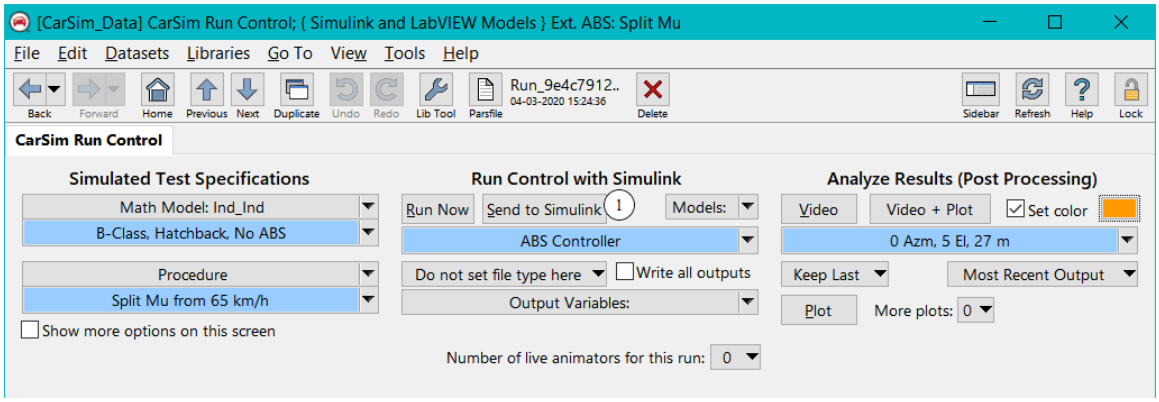

Figure 4. The Run Control dataset.

2. Setup or Verify the C compiler for Simulink Coder by typing "mex -setup" in the Matlab Command Window (Figure 6).
3. In Simulink, open the model Settings from **Simulation > Model Configuration Parameters** (or shortcut Ctrl+E) (Figure 7). Edit settings in the following submenus:

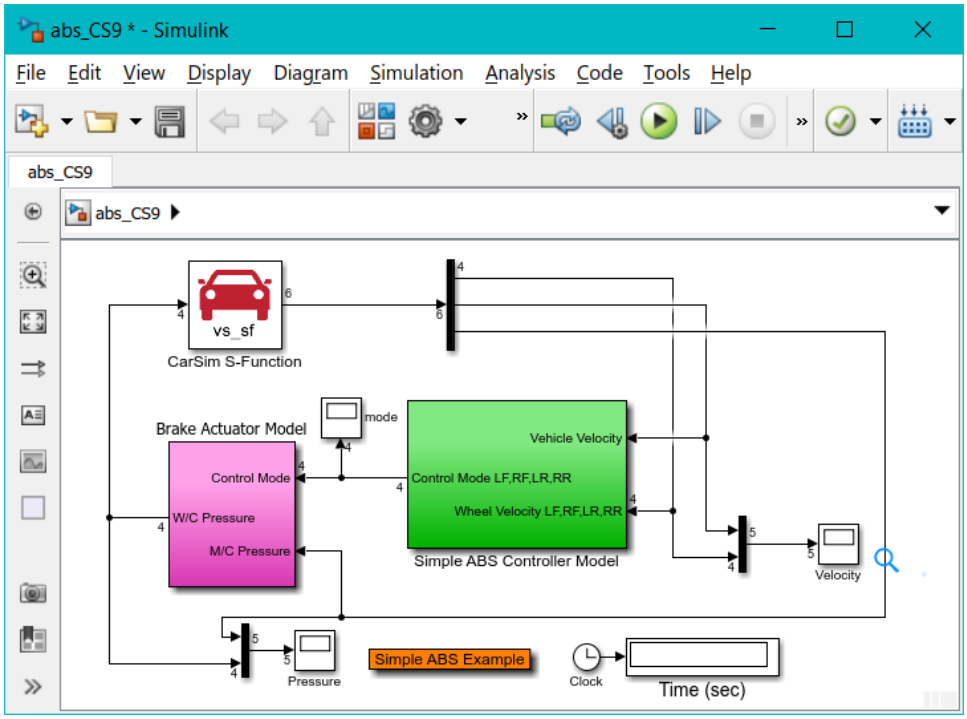

Figure 5. Simulink model with connections to the CarSim S-Function.

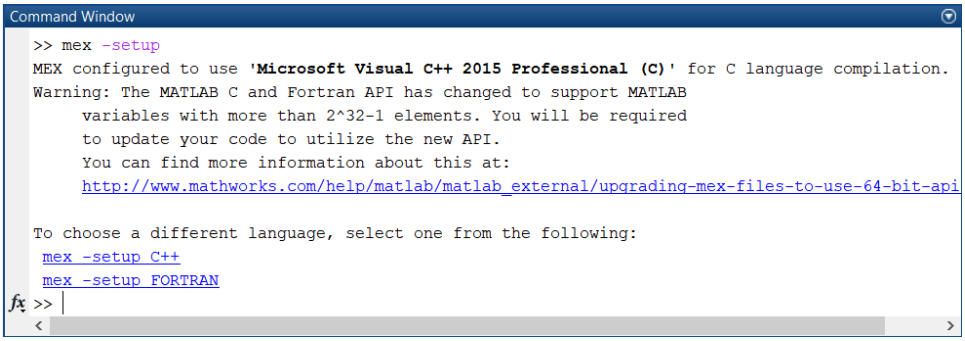

Figure 6. Setup C Compiler for Matlab/Simulink Coder.

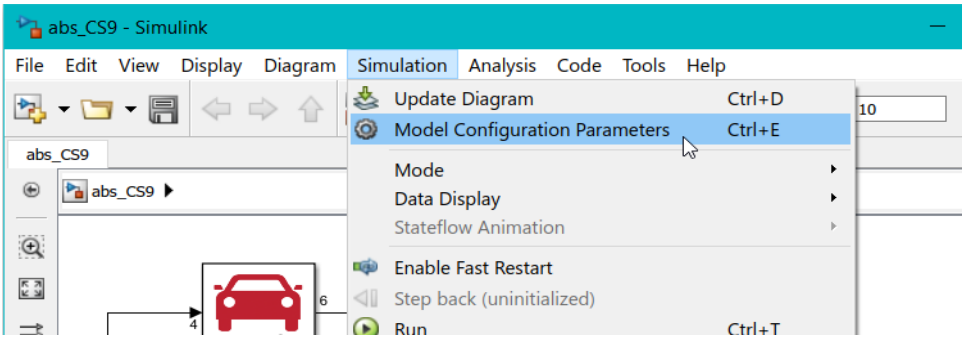

Figure 7. Open Model Configuration Parameters.

a. **Solver**

i. Solver Type is "Fixed-Step" (1) (Figure 8).

ii. Fixed-Step Size (2) (Figure 8) to match Math model Time step (2) (Figure 3).

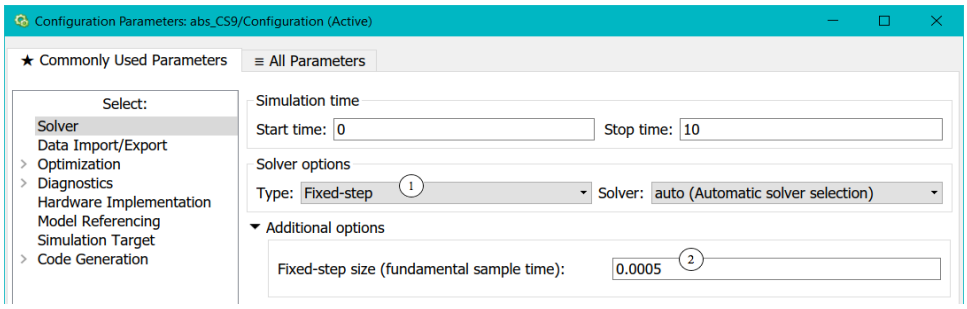

Figure 8. Setup time step.

a. **Code Generation**

iii. Click "Browse…" (1) (Figure 9) to select "Generic Real-Time Target" (2).

iv. Check the box for "Generate makefile" (3) .

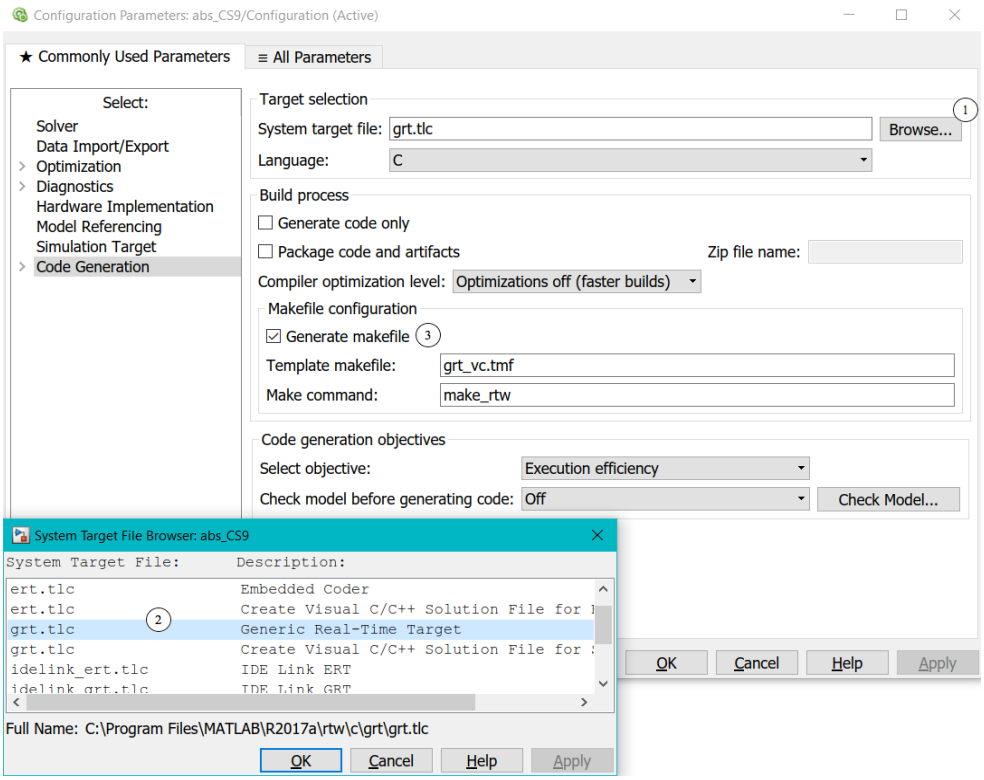

Figure 9. Select Code Generation.

4. Click 'OK'. Navigate back to the Simulink model.

5. Build application from menu **Code > C/C++ Code > Build Model** (or shortcut "Ctrl+B").

6. If there was no error, you should get the file `abs_CS9.exe`.
7. Open a command Window and go to the Simulink model folder. Type `abs_CS9` to run the executable model (Figure 10).
8. You can plot and animate the results from the VehicleSim product GUI.

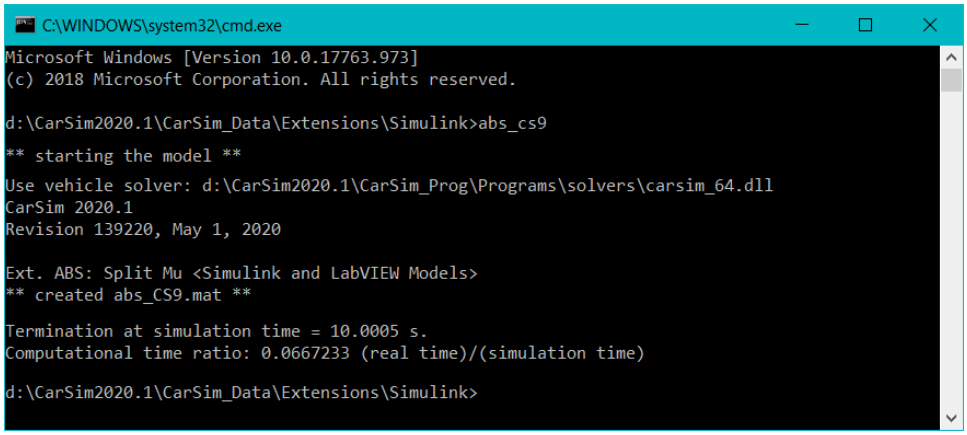

Figure 10. Run the Simulink Model executable file.

Run the Application Without the Full Installed Dataset

In order to run the executable file at other locations or computers, the parsfiles/simfile must be manually copied and the simfile modified.

1. Open the results folder from the Browser menu **View > Open Results Folder for this Run in Windows** (Figure 11).

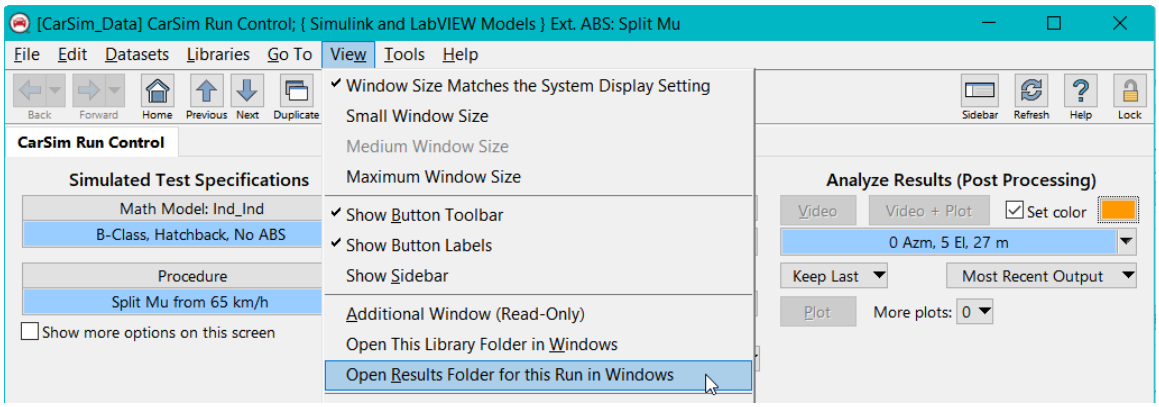

Figure 11. Open Results Folder for this Run.

2. Copy `Run_all.par` from the results folder (Figure 12) to the executable file location.

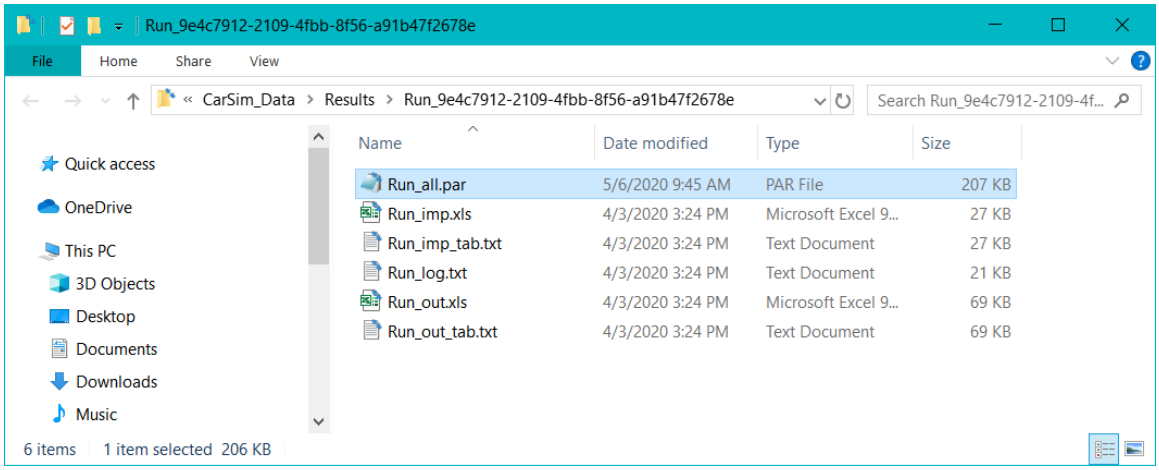

Figure 12. Files in Results Folder.

3. Copy `carsim_64.dll` for 64-bit Simulink or `carsim_32.dll` for 32-bit Simulink from `{CarSim_Prog}\Programs\solvers` to the executable file location.
4. Copy `simfile.sim` from the Simulink model folder to the executable file location.
5. Open the original `simfile.sim` (Figure 13) with a text editor and modify it (Figure 14).
6. Run the executable file and results will generate in your current location.
7. If you have events in your simulation settings, all connected events must be copied to the events folder. This events folder must be at the executable file location.

```
SIMFILE

SET_MACRO $(ROOT_FILE_NAME)$ Run_9e4c7912-2109-4fbb-8f56-a91b47f2678e
SET_MACRO $(OUTPUT_PATH)$ D:\CarSim2020.1\CarSim_Data\Results
SET_MACRO $(WORK_DIR)$ D:\CarSim2020.1\CarSim_Data\
SET_MACRO $(OUTPUT_FILE_PREFIX)$ $(WORK_DIR)$Results\Run_9e4c7912-2109-4fbb-8f56-a91b47f2678e\LastRun

FILEBASE $(OUTPUT_FILE_PREFIX)$
INPUT $(WORK_DIR)$Results\$(ROOT_FILE_NAME)$\Run_all.par
INPUTARCHIVE $(OUTPUT_FILE_PREFIX)$_all.par
ECHO $(OUTPUT_FILE_PREFIX)$_echo.par
FINAL $(OUTPUT_FILE_PREFIX)$_end.par
LOGFILE $(OUTPUT_FILE_PREFIX)$_log.txt
ERDFILE $(OUTPUT_FILE_PREFIX)$.vs
PROGDIR d:\CarSim2020.1\CarSim_Prog\
DATADIR D:\CarSim2020.1\CarSim_Data\
GUI_REFRESH_V CarSim_RefreshEvent_14780
RESOURCEDIR d:\CarSim2020.1\CarSim_Prog\\Resources\
PRODUCT_ID CarSim
PRODUCT_VER 2020.1
ANIFILE D:\CarSim2020.1\CarSim_Data\runs\animator.par
VEHICLE_CODE i_i
EXT_MODEL_STEP 0.00050000
PORTS_IMP 1,4
PORTS_EXP 1,6

END
```

Figure 13. Original simfile.sim generated by VehicleSim Browser.

```
SIMFILE
INPUT Run_all.par
ECHO  Run_echo.par
FINAL Run_end.par
LOGFILE Run_log.txt
ERDFILE Run.vs
PRODUCT_ID CarSim
PRODUCT_VER 2020.1
VEHICLE_CODE i_i
EXT_MODEL_STEP 0.00050000
PORTS_IMP 1,4
PORTS_EXP 1,6
DLLFILE carsim_64.dll
END
```

Figure 14. Modified simfile.sim

Solver Run-time Licenses

The VS Solver libraries are:

CarSim: `carsim_32.dll/carsim_64.dll`

TruckSim: `trucksim_32.dll/trucksim_64.dll`

BikeSim: `bikesim_32.dll/bikesim_64.dll`

Running models made with these solvers requires solver run-time licenses. These licenses are provided by the VehicleSim GUI or by running command-line license control:

CarSim: `cslm.exe`

TruckSim: `tslm.exe`

BikeSim: `bslm.exe`