Mechanical Simulation Technical Memo

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

VehicleSim Dynamics Plugin for Unreal Engine Tutorial

Unreal Engine is a powerful gaming engine which has the capacity to provide high-quality graphics. The Mechanical Simulation **VehicleSim Dynamics plugin for Unreal Engine** allows users to create these high-quality graphics and include **CarSim** and/or **TruckSim** solvers. The plugin will feed terrain data, including surface friction, from Unreal Engine to the CarSim and/or TruckSim math model. It will also provide the math model data to the simulation using C++ code or Blueprints (Unreal Engine's block-based visual scripting system). Standard CarSim output files are generated so that you can view the selected variable outputs in the VS Visualizer plotter after a run. VS Visualizer is included with all VehicleSim products, including CarSim and TruckSim.

This tutorial assumes you have little or no experience with Unreal Engine and that you would like to use the plugin for your vehicle simulations.

This tutorial focuses on using CarSim with the VehicleSim plugin for Unreal, but TruckSim may also be used with the plugin.

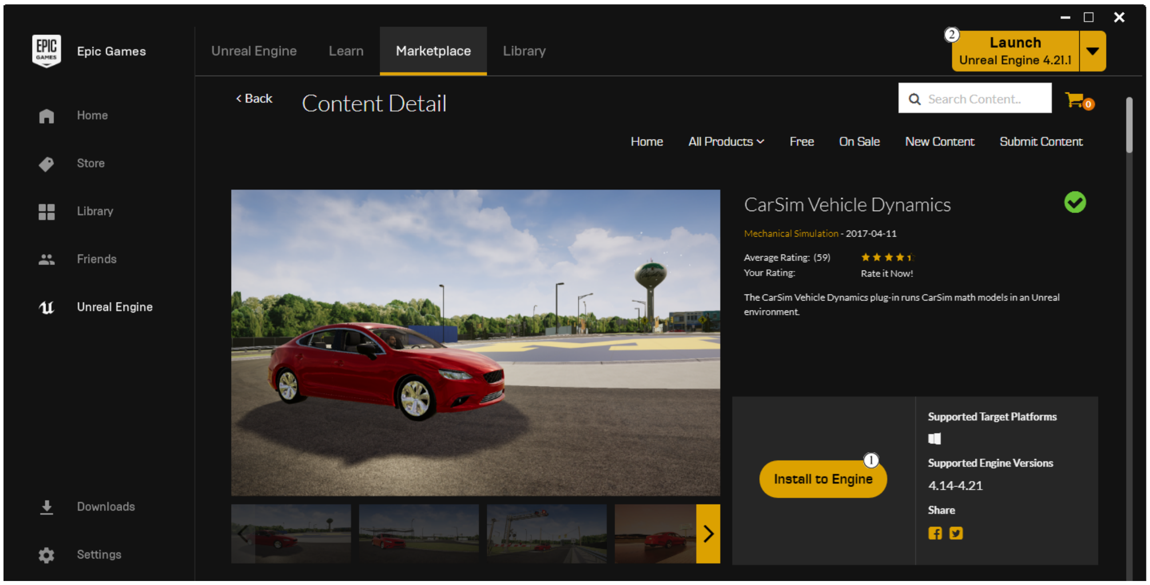
Before You Start

Before starting, there are a few requirements.

1. CarSim 2019.0 must be installed on your computer.
2. You must have a valid CarSim 2019.0 license. Ensure that either the main CarSim program or the CLSM license utility is running with the CarSim Solver license feature enabled.
3. Create an Epic Games account to download Unreal Engine.
4. Unreal Engine version 4.21 must be installed from www.unrealengine.com.

How to Setup the VehicleSim Dynamics Plugin

Once you've downloaded and installed Unreal Engine, open the Epic Game Launcher to access the application. Go to Marketplace, and search for "CarSim" to find the plugin (Figure 1). Click **Install to Engine** ①, then click the **Launch** button ② to access the project browser.



CarSim Vehicle Dynamics

Mechanical Simulation - 2017-04-11

Average Rating: (59) ★★★★½

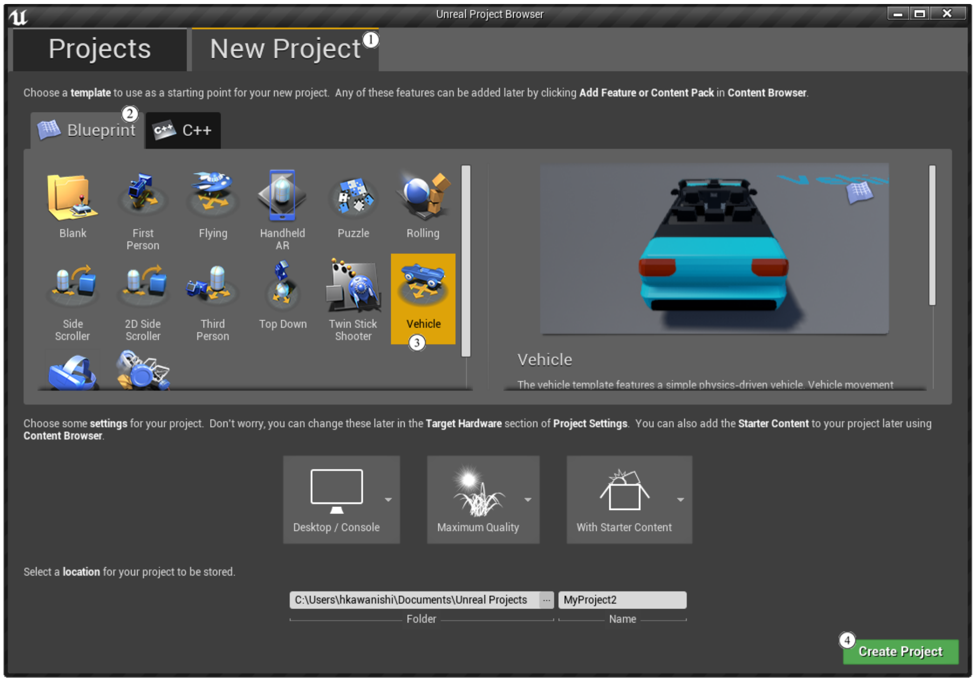
Your Rating: Rate it Now!

The CarSim Vehicle Dynamics plug-in runs CarSim math models in an Unreal environment.

Supported Engine Versions: 4.14-4.21

Figure 1. VehicleSim Dynamics plugin installation.

In the browser (Figure 2), go to the tab labeled **New Project** ①, and from there, navigate to the **Blueprint** tab ② (you can create a C++ project if you prefer). Select **Vehicle** from the list of options ③, then click the **Create Project** button in the lower right corner ④.



Choose a **template** to use as a starting point for your new project. Any of these features can be added later by clicking **Add Feature or Content Pack** in **Content Browser**.

The vehicle template features a simple physics-driven vehicle. Vehicle movement

Choose some **settings** for your project. Don't worry, you can change these later in the **Target Hardware** section of **Project Settings**. You can also add the **Starter Content** to your project later using **Content Browser**.

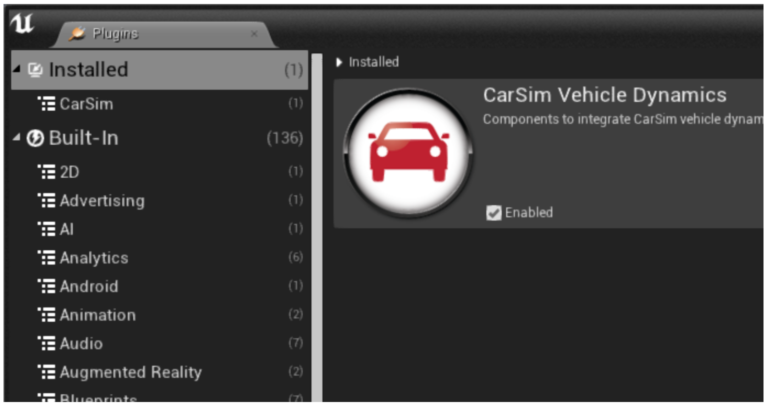
Select a **location** for your project to be stored.

C:\Users\hkawanishi\Documents\Unreal Projects

MyProject2

Figure 2. Unreal project browser.

Note The VehicleSim Dynamics plugin must first be enabled through the Unreal browser before you have access to it. If you are not prompted by the engine to do so, you can navigate from the Unreal browser to **Edit > Plugins**, where you should click the **Enable** checkbox (Figure 3). You will then be prompted to restart Unreal; please follow the prompt and restart.



CarSim Vehicle Dynamics

Components to integrate CarSim vehicle dynam

☑ Enabled

Figure 3. Plugin Browser Screen.

This creates a new project using the Blueprint **Vehicle Template Project**. Once you've created the project, navigate to the left side of the screen under **Modes** ① (Figure 4) and search for

CarSim ②. An item named **Carsim Vehicle Pawn** will show up under the search bar. You have now located the proper plugin component we will be using with the Unreal Plugin.

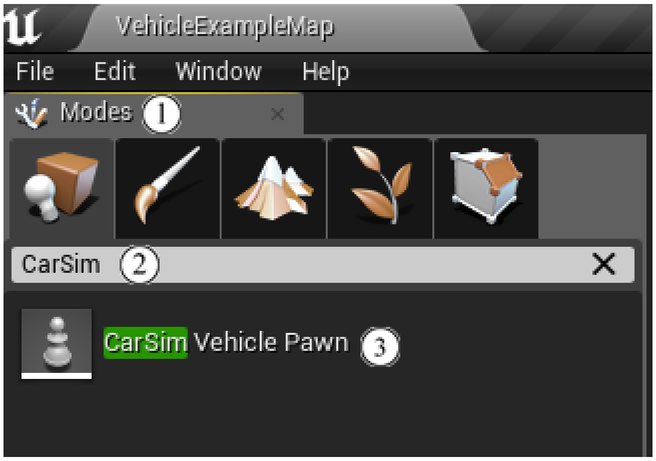
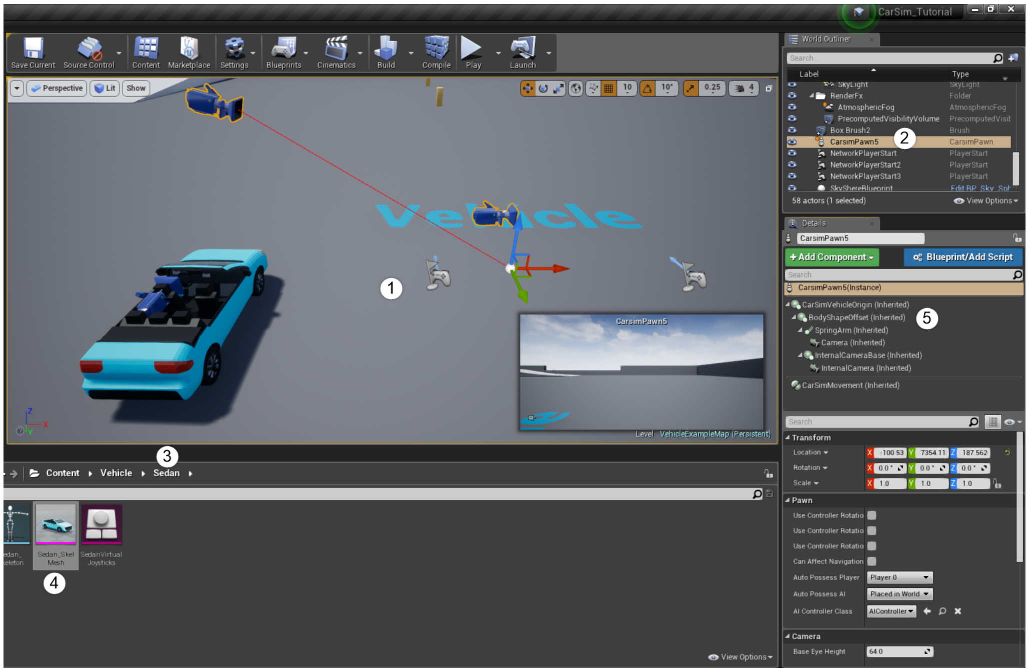


Figure 4. Finding the CarSim Pawn.

How to Attach Objects to the Simulation

In the following steps, we will add a body object to the project, attach a body mesh to it, and finally adjust the offset coordinates such that the overlay matches the origin of the CarSim coordinate system.

1. Drag the **CarSim Vehicle Pawn** to the Unreal Viewport ① (Figure 5). We have now located the correct class object to use with the plugin, and have dragged it into the window from the menu.



Location X -100.53 Y 7354.11 Z 187.562
Rotation X 0.0° Y 0.0° Z 0.0°
Scale X 1.0 Y 1.0 Z 1.0

Use Controller Rotatio □
Use Controller Rotatio □
Use Controller Rotatio □
Can Affect Navigation □
Auto Possess Player: Player 0
Auto Possess AI: Placed in World
AI Controller Class: AIController
Base Eye Height: 64.0

Figure 5. Navigating the Unreal GUI.

> **Note** The vehicle will not have any body geometry attached initially. It will appear as an xyz origin with cameras, as shown above.

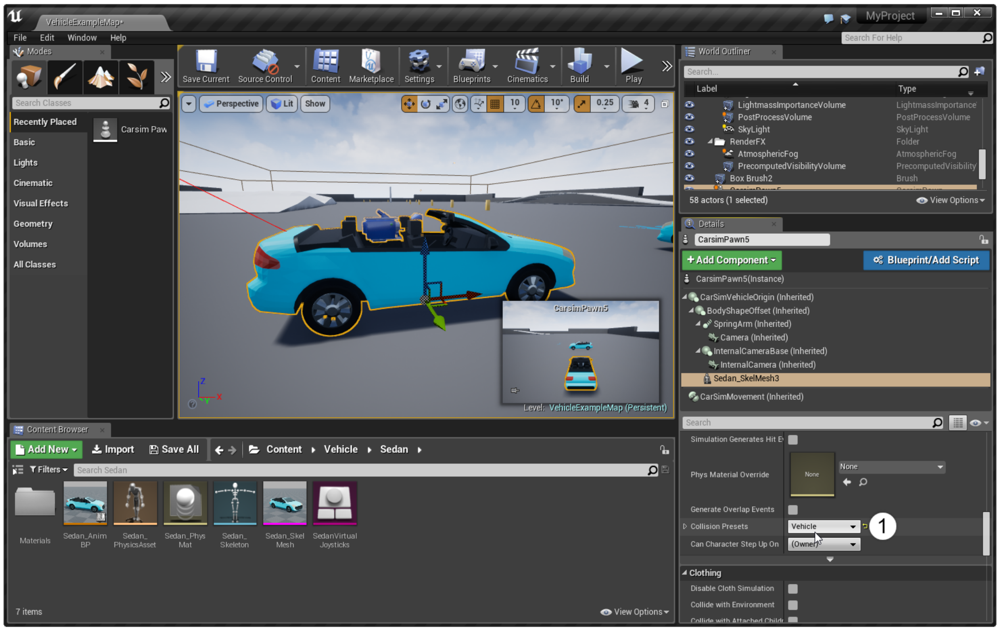
2. Toward the top right side of the Unreal Editor screen is a menu box called the **World Outliner** ②. Locate the **CarSimPawn** and highlight by left-clicking once.

3. At the lower part of the screen is the project space. Navigate to the example vehicle body in the content browser **Content > Vehicle > Sedan** ③ and drag **Sedan_SkelMesh** ④ to the **Details** menu, dropping the mesh onto **BodyShapeOffset** ⑤. It will prompt you for a name, just press enter to save as **Sedan_SkelMesh**.

4. This will attach the body mesh to the **CarSimPawn** object.

5. Navigate to the **Sedan_SkelMesh** details ① (Figure 6). Here, you will find physics and collision elements to manage. Scroll down until you find the **Collision Presets**, and use the drop-down menu to change collision preset to **Vehicle** on the body mesh.

Note Both the vehicle from the example project (**Sedan**) and the CarSim vehicle have their Auto Possess Player option set to Player 0, which may cause some confusion if you're not aware of this naming convention. To avoid this, set Auto Possess Player to **Disable** for the **Sedan**.



Label	Type
LightmassImportanceVolume	LightmassImportance
PostProcessVolume	PostProcessVolume
SkyLight	SkyLight
RenderFX	Folder
AtmosphericFog	AtmosphericFog
PrecomputedVisibilityVolume	PrecomputedVisibility
Box Brush2	Brush

CarsimPawn5
CarsimPawn5(Instance)
CarSimVehicleOrigin (Inherited)
BodyShapeOffset (Inherited)
SpringArm (Inherited)
Camera (Inherited)
InternalCameraBase (Inherited)
InternalCamera (Inherited)
Sedan_SkelMesh3
CarSimMovement (Inherited)

Simulation Generates Hit E ☐
Phys Material Override: None
Generate Overlap Events ☐
Collision Presets: Vehicle
Can Character Step Up On: (Owner)
Clothing
Disable Cloth Simulation ☐
Collide with Environment ☐
Collide with Attached Childr ☐

Figure 6. Set the collision presets to vehicle.

Hit the **Play** button on the top of the Unreal Editor to drive the CarSim vehicle using the existing control mapping from the **Sedan** project (Figure 6) and click the **Viewport**. If a keyboard is used, the vehicle can be maneuvered with the WADS or arrow keys. Alternatively, you could also operate a gamepad or a gaming steering wheel kit to drive the vehicle.

After stopping the simulation (if a mouse cursor is not visible, hit "Escape"), the CarSim output files can be found in:

User home folder > **AppData** > **Local** > **UnrealEngine** > **Common** > **CarSim_plugin**

Note that the **AppData** folder is often treated as a Hidden Item in Windows Explorer. Click the checkbox to show **Hidden Items** under the **VIEW** tab of the explorer window.

Loading an Example Project

We have tested this example using CarSim 2019.0 and Unreal Engine version 4.21.

1. From the Unreal Marketplace, search CarSim. Click CarSim Vehicle Dynamics plug in icon (Figure 1. VehicleSim Dynamics plugin installation.). The link to the Example Project zip file is found at the bottom of the **Technical Details** section. Download the zip file and unzip into **Unreal Project** folder.

Note You can save the Project to any directory. However, if it is saved in the Unreal Project folder, it will automatically show up in the **Unreal Project Browser**.

2. Launch Unreal Engine.
3. In the Unreal Project Browser, select **Example_UE4_Project.uproject** file. If this example project file was not saved in Unreal Project folder, browse and find the project file.

Note If a message appears saying modules are missing or requesting to build with a different engine, click **Yes**. This occurs when a newer version of Unreal Engine is used.

4. The Unreal Editor will open the project with two vehicles in a driving environment (Figure 7).

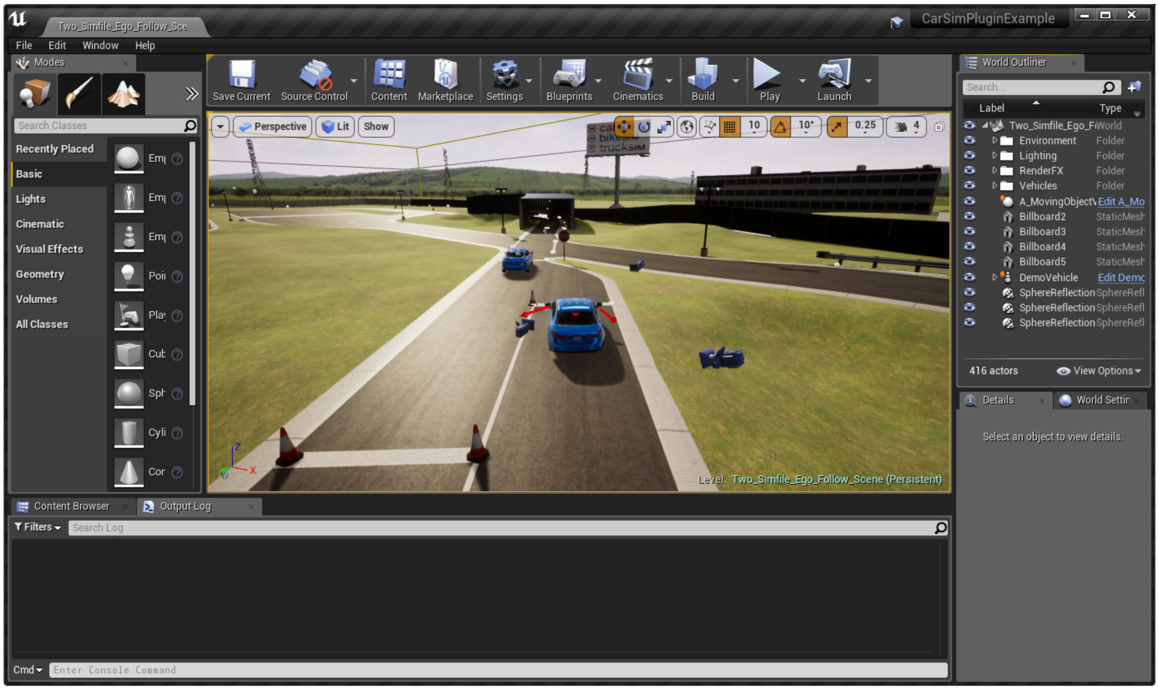


Figure 7. Example project.

5. The VehicleSim Dynamics plugin requires a CarSim license to run. To enable the CarSim license, ensure that either:

   a. The main CarSim program, also known as **VS Browser,** is running.

   OR

   b. The VLSM license utility is running with the CarSim Solver license feature enabled.

6. Hit the **Play** button on the top of Unreal Engine to drive the CarSim vehicle using the existing control mapping form the example project. The vehicle can be moved using a keyboard with WASD or arrow keys. A gamepad may also be used.

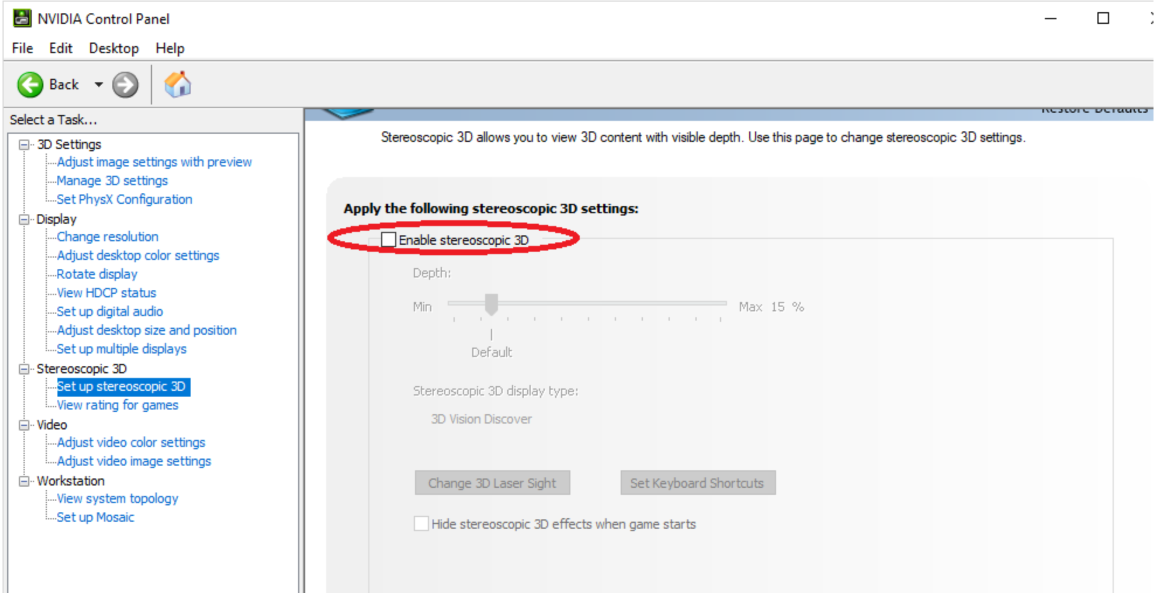
> **Note** The Marketplace project contains two vehicles in order to support the VS Connect example. When testing a single vehicle as outlined in this document, you will receive multiple warnings indicating “Unable to locate VS Solver Variable”. These can be safely ignored. The second vehicle, named **A_MovingObjectVehicle** in the **World Outliner**, will set itself to invisible when **Play** is hit if it is not being updated via VS Connect. For more details about using VS Connect with The CarSim Unreal Plugin, see:
>
> http://www.unrealengine.com/marketplace/carsim-vehicle-dynamics

If a user-specified CarSim model is not provided, Unreal Engine uses a default CarSim model.

The default CarSim model is included in the VehicleSim Dynamics Plugin directory, and it is found in:

Unreal Engine plugin directory: Plugins > Marketplace > CarSim > ThirdParty > CarSim

> **Note** When you hit the Play button, and the animation looks like it in 3D mode, you need to turn off Stereoscopic 3D Mode in NVIDIA control panel (Figure 8).



Stereoscopic 3D allows you to view 3D content with visible depth. Use this page to change stereoscopic 3D settings.

Apply the following stereoscopic 3D settings:

□ Enable stereoscopic 3D

Stereoscopic 3D display type:

3D Vision Discover

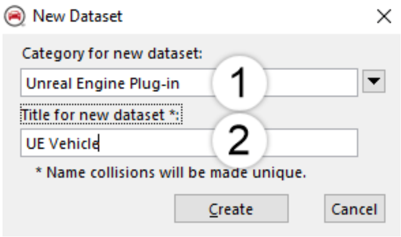
□ Hide stereoscopic 3D effects when game starts

Figure 8. Disabling Stereoscopic 3D Mode in NVIDIA control panel.

Creating a Vehicle Model in CarSim

As mentioned in the previous section, Unreal Engine uses a default VehicleSim Dynamics Plugin model if a vehicle model is not specified. This section describes how to set a CarSim model and use it to run inside Unreal Engine.

Open CarSim. Navigate to **File > New Dataset (Empty)** to create a new dataset in the Run Control Library. In **Category** for the new dataset, enter **Unreal Engine Plug-in** ①, and in **Title** for the new dataset, enter **UE Vehicle** ② (Figure 9).



Category for new dataset:

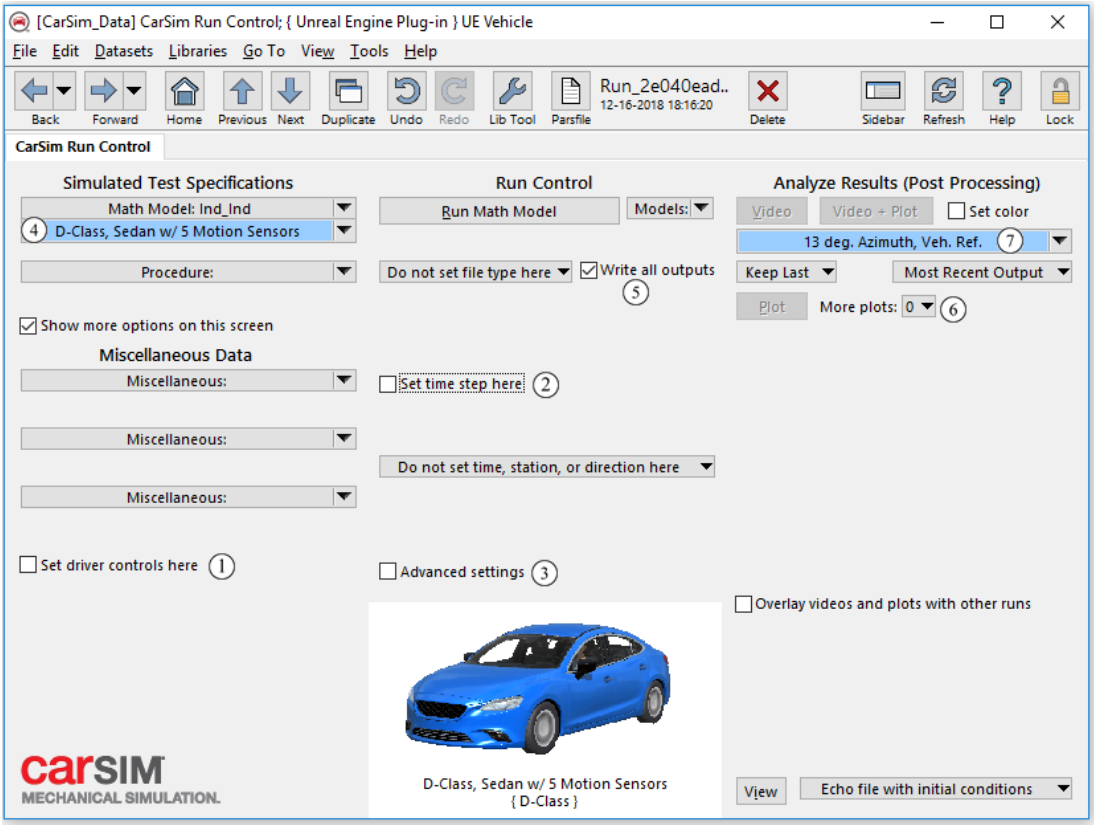
Unreal Engine Plug-in

Title for new dataset *:

UE Vehicle

* Name collisions will be made unique.

Figure 9. Creating a new CarSim dataset.



[CarSim_Data] CarSim Run Control; { Unreal Engine Plug-in } UE Vehicle

Run_2e040ead..
12-16-2018 18:16:20

CarSim Run Control

Simulated Test Specifications

Math Model: Ind_Ind

④ D-Class, Sedan w/ 5 Motion Sensors

Procedure:

☑ Show more options on this screen

Miscellaneous Data

Miscellaneous:

Miscellaneous:

Miscellaneous:

□ Set driver controls here ①

Run Control

Run Math Model

Models:

Do not set file type here

☑ Write all outputs ⑤

□ Set time step here ②

Do not set time, station, or direction here

□ Advanced settings ③

D-Class, Sedan w/ 5 Motion Sensors
{ D-Class }

Analyze Results (Post Processing)

Video

Video + Plot

□ Set color

13 deg. Azimuth, Veh. Ref. ⑦

Keep Last

Most Recent Output

Plot

More plots: 0 ⑥

□ Overlay videos and plots with other runs

View

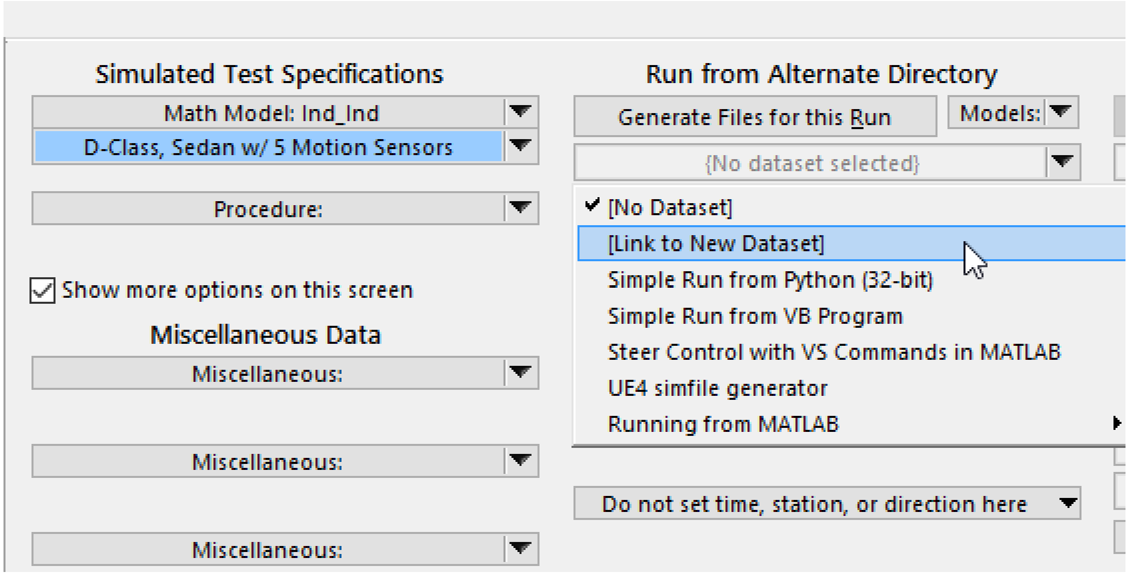
Echo file with initial conditions

Figure 10. Selecting a vehicle in a new dataset.

In this new dataset (Figure 10), uncheck **Set driver controls here** ①, **Set time step here** ②, and **Advanced settings** ③. Use the pull-down menu next to **Math Model** and select **Vehicles** > **Vehicle with Loads, Sensors, Trailers, etc.** Then, select **D-Class, Sedan w/ 5 Motion Sensors**④. Check **Write all outputs** ⑤. Select 0 from the **More plots** drop down menu ⑥. Select **Vehicle Reference** > **13 deg. Azimuth, Veh. Ref.** for Animator: Camera Setup dataset ⑦.

Note Currently, the braking can only be controlled by the pedal force using Unreal Engine plug-in. You need to select Control with pedal force option in the brake dataset.

Next, we will create another dataset which will be used to generate the simfile to connect the vehicle dynamics to the plugin. After setting up the vehicle model, select **Models: Transfer to Local Windows Directory** from the **Models** drop-down menu. Select [Link to New Dataset] (Figure 11). In **Title** for the new dataset, enter **UE4 simfile generator**.


Simulated Test Specifications
Procedure:
Miscellaneous Data
Miscellaneous:
Miscellaneous:
Miscellaneous:
Run from Alternate Directory
Generate Files for this Run
Models:
{No dataset selected}
[No Dataset]
[Link to New Dataset]
Simple Run from Python (32-bit)
Simple Run from VB Program
Steer Control with VS Commands in MATLAB
UE4 simfile generator
Running from MATLAB

Math Model: Ind_Ind

D-Class, Sedan w/ 5 Motion Sensors

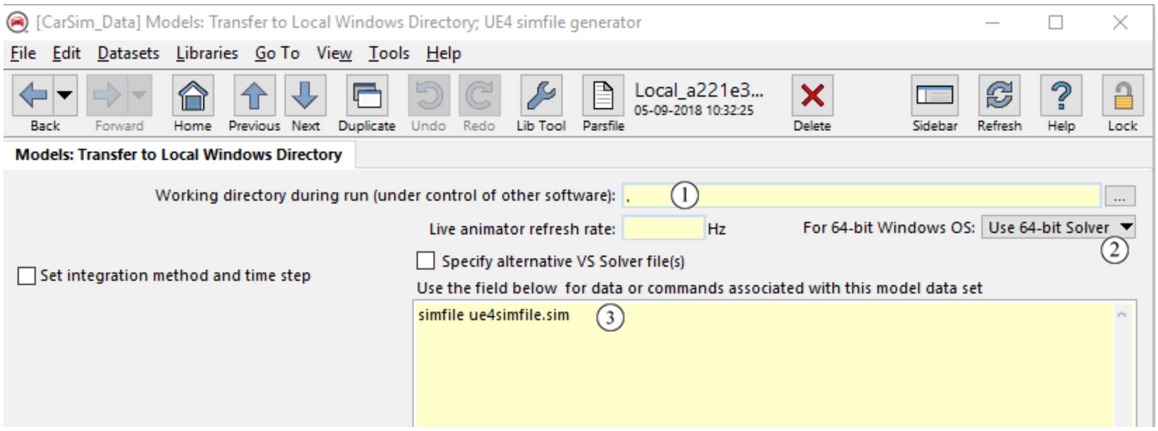
☑ Show more options on this screen

Do not set time, station, or direction here

Figure 11. Select Models: Transfer to Local Windows Directory and select [Link to New Dataset]

Click the blue link to enter to this newly created **UE4 simfile generator** dataset (Note: This **UE4 simfile generator** dataset is provided in CarSim 2019.0).

UE4 simfile generator dataset should contain the information shown in Figure 12. The working directory should contain a period, "." ①. When there is a period, ".", in this field, CarSim will create a simfile in the current database directory. Select to use 64-bit solver ②. Write `simfile ue4simfile.sim` in the yellow field ③. In this example, the simfile name is specified as ue4simfile.sim ③.


[CarSim_Data] Models: Transfer to Local Windows Directory; UE4 simfile generator
File Edit Datasets Libraries Go To View Tools Help
Back
Forward
Home
Previous
Next
Duplicate
Undo
Redo
Lib Tool
Parsfile
Local_a221e3...
05-09-2018 10:32:25
Delete
Sidebar
Refresh
Help
Lock
Models: Transfer to Local Windows Directory
1
2
3

Working directory during run (under control of other software): .

Live animator refresh rate: Hz

For 64-bit Windows OS: Use 64-bit Solver

□ Set integration method and time step

□ Specify alternative VS Solver file(s)

Use the field below for data or commands associated with this model data set

```
simfile ue4simfile.sim
```

Figure 12. UE4 simfile generator dataset Models screen.

Next, go back to the **Run Control** screen, and click **Generate Files for This Run** button. Then, navigate to the working directory you just set up (the current database), locate the **ue4simfile.sim** you just created, and press **Shift + Right-Click** on it. Select **Copy as Path** from the resulting dropdown-menu.

Go back to the Unreal Editor. Find **MSC_DemoVehicle** in **World Outliner** and right-click to highlight it. Then, in the **Details** tab, scroll down to find and click **CarSimMovement (Inherited)** to highlight the component ① (Figure 13).

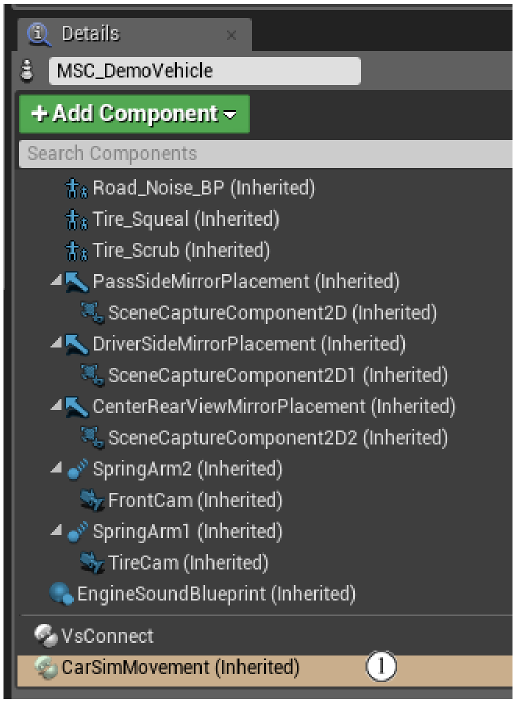
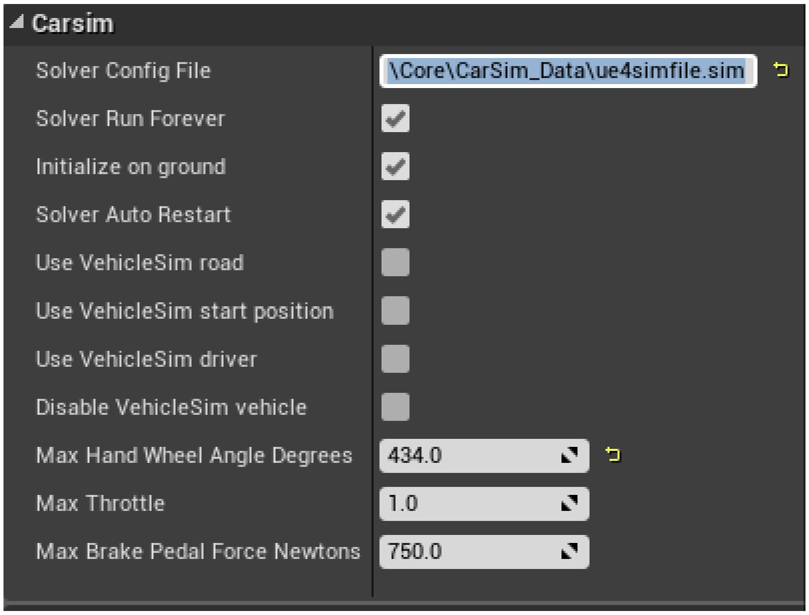

Details
MSC_DemoVehicle
+ Add Component
Search Components
Road_Noise_BP (Inherited)
Tire_Squeal (Inherited)
Tire_Scrub (Inherited)
PassSideMirrorPlacement (Inherited)
SceneCaptureComponent2D (Inherited)
DriverSideMirrorPlacement (Inherited)
SceneCaptureComponent2D1 (Inherited)
CenterRearViewMirrorPlacement (Inherited)
SceneCaptureComponent2D2 (Inherited)
SpringArm2 (Inherited)
FrontCam (Inherited)
SpringArm1 (Inherited)
TireCam (Inherited)
EngineSoundBlueprint (Inherited)
VsConnect
CarSimMovement (Inherited)
1

Figure 13. CarSim movement using Unreal.

In the Unreal Editor, still highlighting **CarSimMovement (Inherited)** component, scroll down to find **CarSim** > **Solver Config File** and **Paste** the path you copied in the previous step (Figure 14).

Note The path created using Copy Path in Windows Explorer includes a quotation mark, " , at the beginning and at the end of the path. Delete these two quotation marks in the pathname.



Carsim

Solver Config File	\Core\CarSim_Data\ue4simfile.sim
Solver Run Forever	☑
Initialize on ground	☑
Solver Auto Restart	☑
Use VehicleSim road	□
Use VehicleSim start position	□
Use VehicleSim driver	□
Disable VehicleSim vehicle	□
Max Hand Wheel Angle Degrees	434.0
Max Throttle	1.0
Max Brake Pedal Force Newtons	750.0

Figure 14. Vs Config File Setup.

Press the **Play** button in the Unreal Editor and then click the **ViewPort**. CarSim will now run with the given parameters. The vehicle can be moved using a keyboard with WASD or arrow keys. A gamepad is also used. Press **H** key to display a speedmeter and a tachometer. Press **Tab** key to toggle the camera view between the driver view and the outside view. After stopping the simulation (if a mouse cursor is not visible, hit “Escape”), the CarSim output files can be found from CarSim GUI. Go to **View** > **Open Results Folder for this Run in Windows**.

Changing CarSim Vehicle Parameters

You can now modify the vehicle inside CarSim and test it using the Unreal Editor.

1. From the Run Control screen in CarSim, click the arrow next to the vehicle dataset and select **[Copy and Link Dataset]** to duplicate it. Leave the category name and the title name as the default (Figure 15).

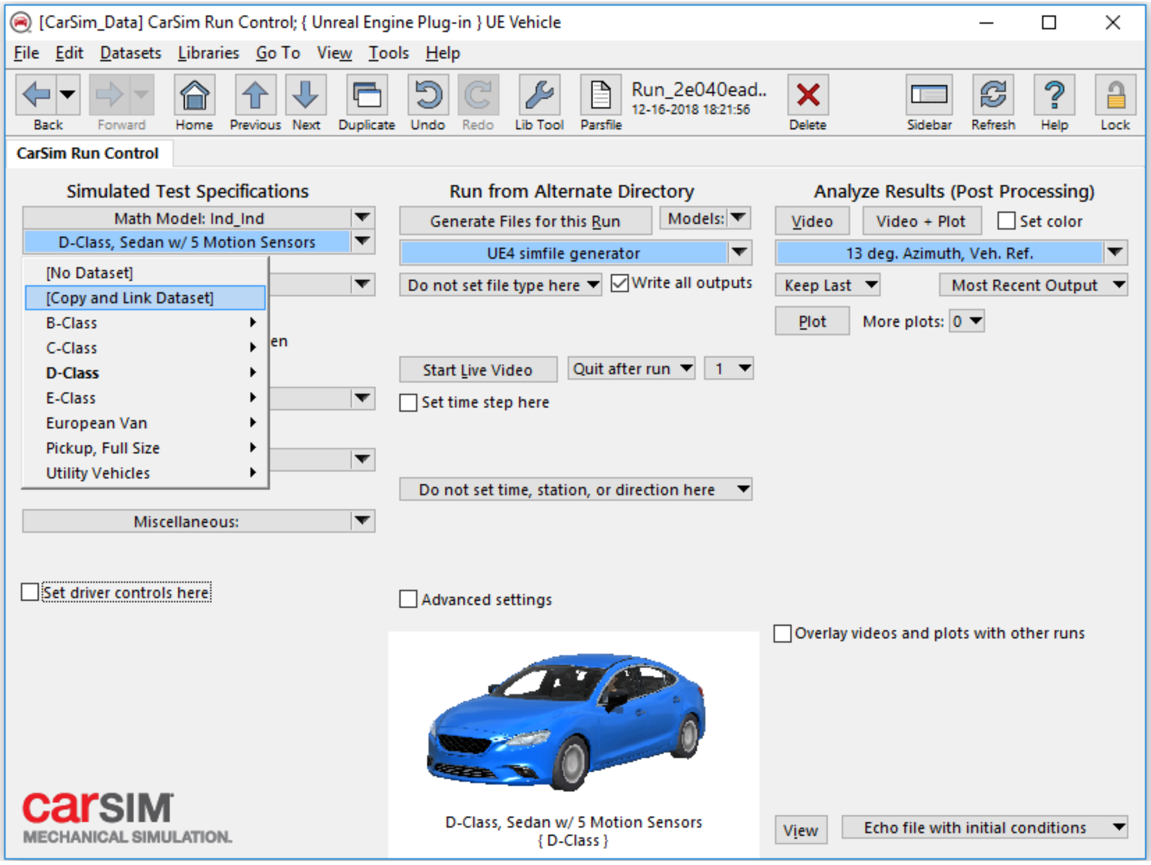


Figure 15. Duplicate Dataset in CarSim.

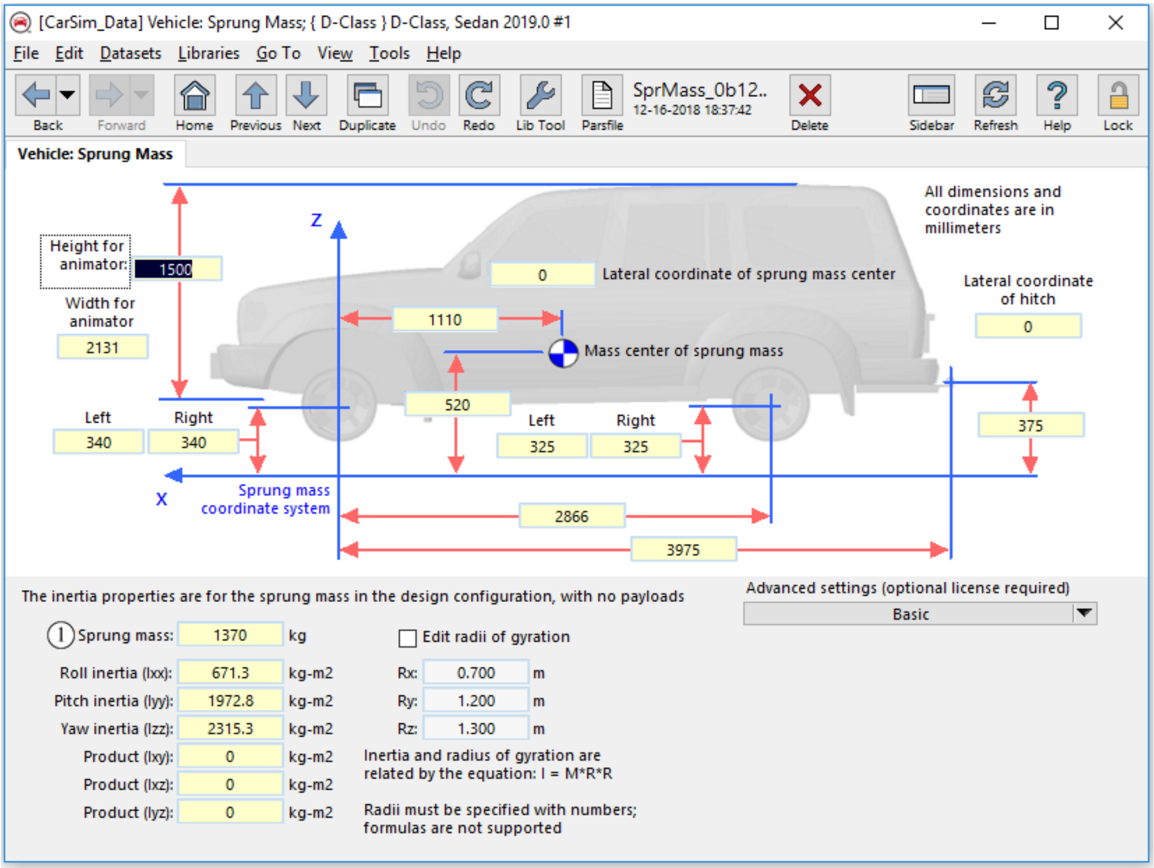
2. Click the blue link **D-Class, Sedan w/ 5 Motion Sensors #1**, select **[Copy and Link Dataset]** to duplicate the **Lead Unit: Ind_Ind** dataset.

    a. Name the title **D-Class, Sedan 2019.0 #1** and click the link.

    b. Repeat the same thing and duplicate the **Rigid Sprung Mass** dataset, giving it the new title **D-Class 2019.0 Sedan #1**.

3. Navigate to the sprung mass dataset by clicking the blue link again, and change the sprung mass value from 1370 Kg (① Figure 16) to 1600 Kg.

4. Click the **Back** button to get back to the **Vehicle: Assembly** screen. Change the tires to **Touring Tires: 265/70 R17**.



All dimensions and coordinates are in millimeters

Height for animator: 1500

Width for animator: 2131

Lateral coordinate of sprung mass center: 0

Lateral coordinate of hitch: 0

1110

520

375

2866

3975

	Left	Right
	340	340
	325	325

The inertia properties are for the sprung mass in the design configuration, with no payloads

Advanced settings (optional license required): Basic

Sprung mass:	1370	kg	
Roll inertia (Ixx):	671.3	kg-m2	
Pitch inertia (Iyy):	1972.8	kg-m2	
Yaw inertia (Izz):	2315.3	kg-m2	
Product (Ixy):	0	kg-m2	
Product (Ixz):	0	kg-m2	
Product (Iyz):	0	kg-m2	

□ Edit radii of gyration

Rx:	0.700	m
Ry:	1.200	m
Rz:	1.300	m

Inertia and radius of gyration are related by the equation: I = M*R*R

Radii must be specified with numbers; formulas are not supported

Figure 16. Changing the vehicle sprung mass in CarSim.

5. Click the **Home** button in CarSim to get back to the Run Control Screen, and then click **Generate Files for this Run** to update ue4simfile.sim.

6. Go back to the Unreal Editor, and press the **Play** button again. This time the vehicle runs with the updated vehicle model with the new sprung mass. Note that since you have not changed the skin in UE4, the vehicle animation in UE4 looks same as before.

The CarSim output files can be found in Results/ folder.

Selecting a Vehicle with a different type of Suspension

It is possible to run a different suspension vehicle using the above example. The instruction in this section is also applied for the front/rear independent suspension vehicle.

1. Go to CarSim Run Control, and change the vehicle type by selecting **Vehicles** > **Vehicle with Loads, Sensors, Trailer, etc.**, and then select from that list **Pickup, Full Size w/ 2000kg Payload**, which has a rear solid axle suspension (Ind_SA) (Figure 17).

Note Currently, the braking can only be controlled by the pedal force using Unreal Engine plug-in. You need to select Control with pedal force option in the brake dataset.

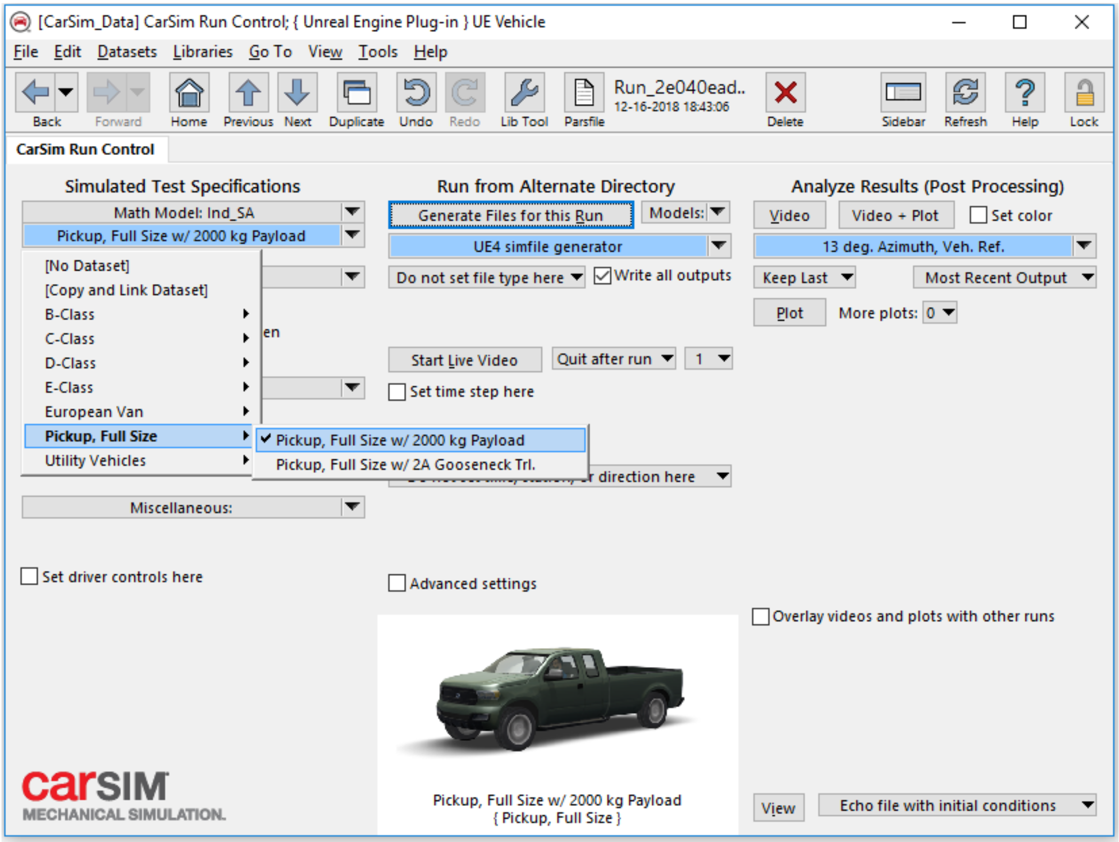


Figure 17. Selecting a different vehicle and suspension.

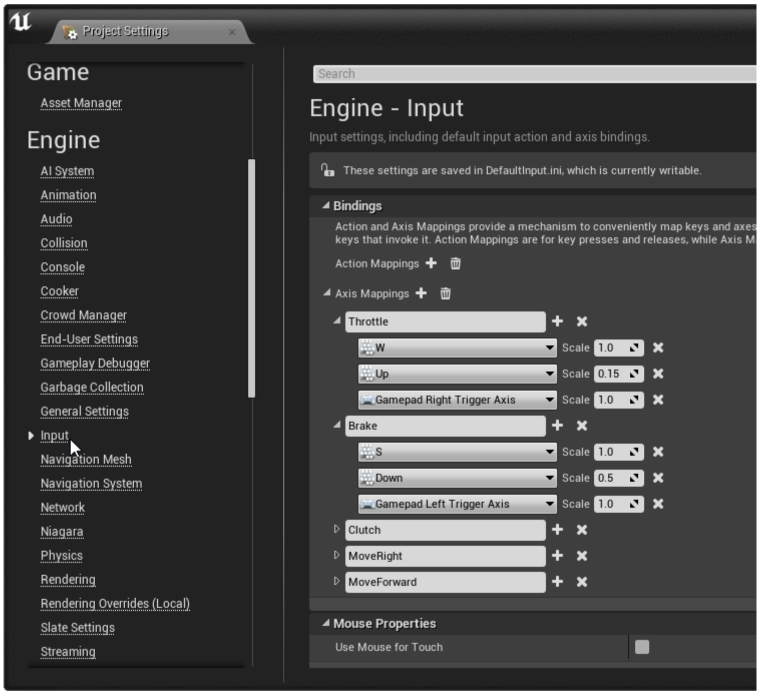
2. Press the **Generate Files for this Run** button. This will update ue4simfile.sim.
3. Go back to the Unreal Editor, and press the **Play** button again and click the **Viewport**. The vehicle will now run with the updated vehicle model. Note that the vehicle animation in UE4 looks same unless you update the skin in UE4.

The results of this run can be found in your CarSim Database results directory. In CarSim, navigate to **View** > **Open Results Folder for this Run in Windows** to find the results.

Input and Configuration Settings of the Unreal VehicleSim Dynamics Plugin

You can adjust the keyboard settings using Settings option in the Unreal Editor.

Go to **Settings** > **Projects Settings**. Select **Input** under **Engine** category (Figure 18). Here, you can set the different keyboard control and the scale of the throttle and the brake. In Figure 18, it is showing that the Brake is 1.0 when using “S” and 0.5 but when using “Down” key. That means, the brake force is applied 50% of the force when using “Down” key.



Engine - Input

Input settings, including default input action and axis bindings.

These settings are saved in DefaultInput.ini, which is currently writable.

Bindings

Action and Axis Mappings provide a mechanism to conveniently map keys and axes
keys that invoke it. Action Mappings are for key presses and releases, while Axis M

Action Mappings

Axis Mappings

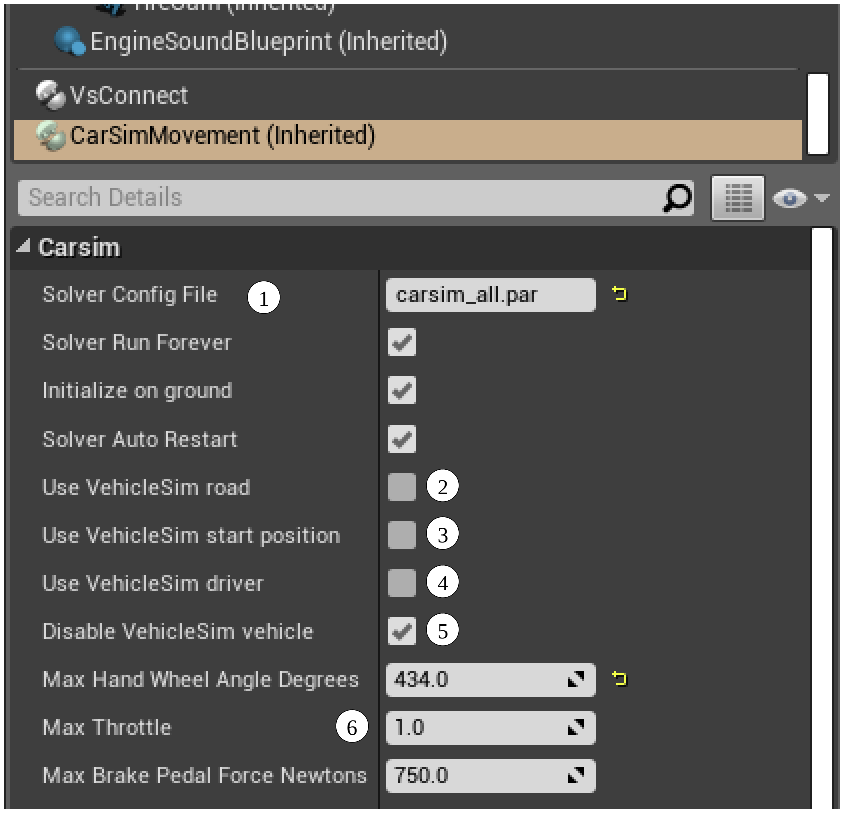
- Throttle
  - W Scale 1.0
  - Up Scale 0.15
  - Gamepad Right Trigger Axis Scale 1.0
- Brake
  - S Scale 1.0
  - Down Scale 0.5
  - Gamepad Left Trigger Axis Scale 1.0
- Clutch
- MoveRight
- MoveForward

Mouse Properties

Use Mouse for Touch □

Figure 18. Unreal Engine Input Settings.

You can select different **CarSim** settings in the **Details** tab as follows (Figure 19).



EngineSoundBlueprint (Inherited)

VsConnect

CarSimMovement (Inherited)

Search Details

Carsim

Solver Config File ①	carsim_all.par
Solver Run Forever	☑
Initialize on ground	☑
Solver Auto Restart	☑
Use VehicleSim road	□ ②
Use VehicleSim start position	□ ③
Use VehicleSim driver	□ ④
Disable VehicleSim vehicle	☑ ⑤
Max Hand Wheel Angle Degrees	434.0
Max Throttle ⑥	1.0
Max Brake Pedal Force Newtons	750.0

Figure 19. CarSim configuration settings.

The **Solver Config File** ① is where you can enter the name of CarSim `*.sim` (Simfile). To use the example models packaged with the plugin, you can simply enter **carsim_all.par** or **trucksim_all.par** depending on which product you're using. If you'd like to inspect these files, they are in your plugin installation location from the Unreal Marketplace. The default location would be **Engine\Plugins\Marketplace\CarSim\ThirdParty\VehicleSim** inside of your engine installation, e.g. **c:\Program Files\Epic Games\UE_4.21**.

Instead of the road in the Unreal scene, you can use the CarSim Road. To do so, select **Use Carsim Road** ②. If this box is checked, the CarSim solver uses the road defined in CarSim. If this box is not checked, the solver will query the Unreal terrain below the vehicle tires. If **Use VehicleSim start position** ③ is checked, the vehicle will be positioned based on the parameters in the Parsfile.

For the driver control, Use **CarSim Driver box** ④ can be checked/unchecked. If the box is checked, CarSim driver model is used and the control is ignored. If unchecked, the vehicle inputs will be passed to the solver but the CarSim driver model will be suppressed. Note that it is important to match the CarSim coordinates and the UE4 coordinates when use CarSim driver model in UE4. For example, if your vehicle's starting position is at (500, 500, 0) in UE4 but the CarSim starts at (0, 0, 0), you are not able to observe a desired maneuver.

If you'd like to keep a vehicle in your scene but not have it activated you can select **Disable VehicleSim vehicle** ⑤. This will prevent the vehicle from doing a license check and loading when the simulation begins. This can be useful for having many vehicles in your scene that you can activate as needed. The example scene will check this flag during **Begin Play** in the blueprints and hide the vehicle from the scene if it is disabled.

If the driver control box is unchecked, **Max Hand Wheel Angle Degrees**, **Max Throttle**, and **Max Brake Pedal Force Newtons** ⑥ can be adjusted.

Changing the Coefficient of Road Friction

If you are not using CarSim Road (② of Figure 19), and if you want to change the road friction coefficient, you can do so inside Unreal Editor.

To change the road friction for the **Example_UE4_Project.uprogject**, in Content Browser, go to **Content** > **Environment** > **Materials** folder (Figure 20).

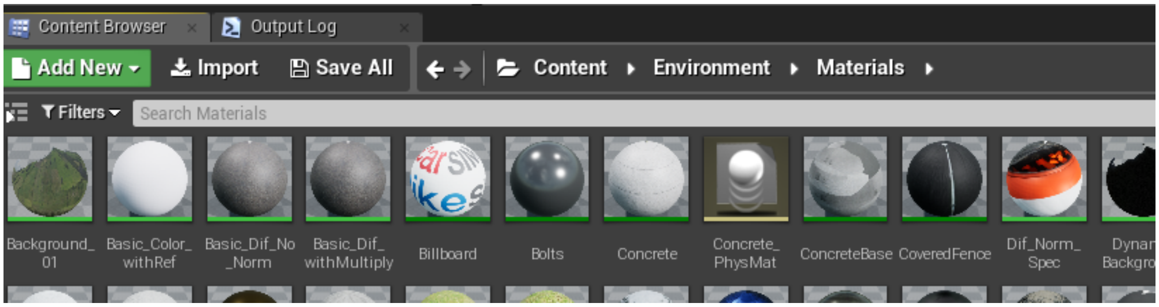


Figure 20. Access to the Materials from Content Browser.

In the right bottom corner of the Content Browser tab, click **View Options**, check **Column** for the View Type, and sort with Type so it is easy to find **Physical Materials** (Figure 21).

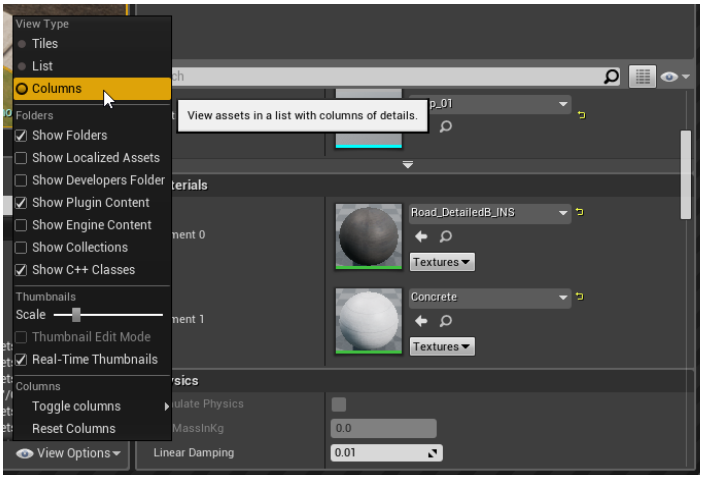
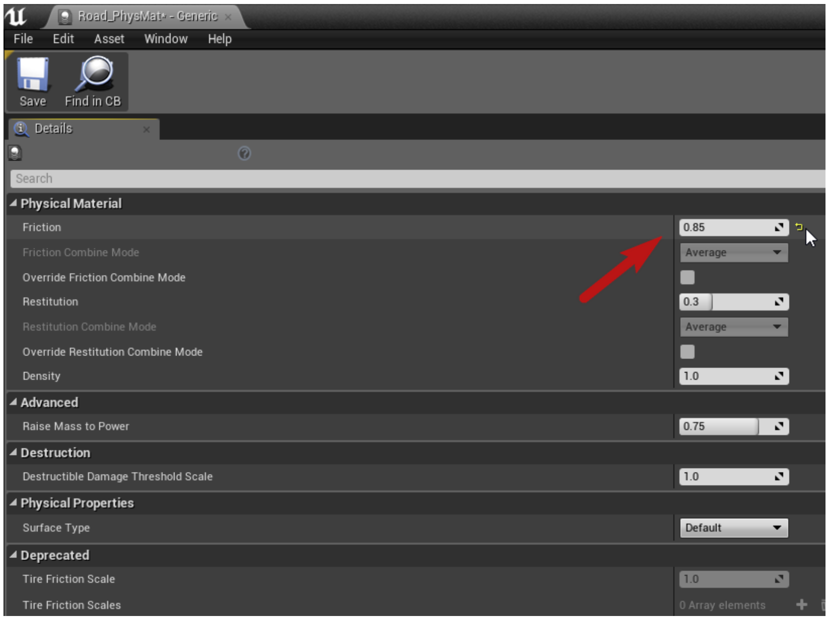


Figure 21. Select Column for the View Options.

In this example, the gray part of the road uses **Road PhysMat** physical material. Click **road_PhysMat**. When the window opens, the **Friction** entry is found under **Physical Material** (Figure 22). Change this value to experiment.

The white part of the road uses **Concrete_PhysMat** material. If necessary, change this friction coefficient as well.



Physical Material	
Friction	0.85
Friction Combine Mode	Average
Override Friction Combine Mode	□
Restitution	0.3
Restitution Combine Mode	Average
Override Restitution Combine Mode	□
Density	1.0
Advanced	
Raise Mass to Power	0.75
Destruction	
Destructible Damage Threshold Scale	1.0
Physical Properties	
Surface Type	Default
Deprecated	
Tire Friction Scale	1.0
Tire Friction Scales	0 Array elements

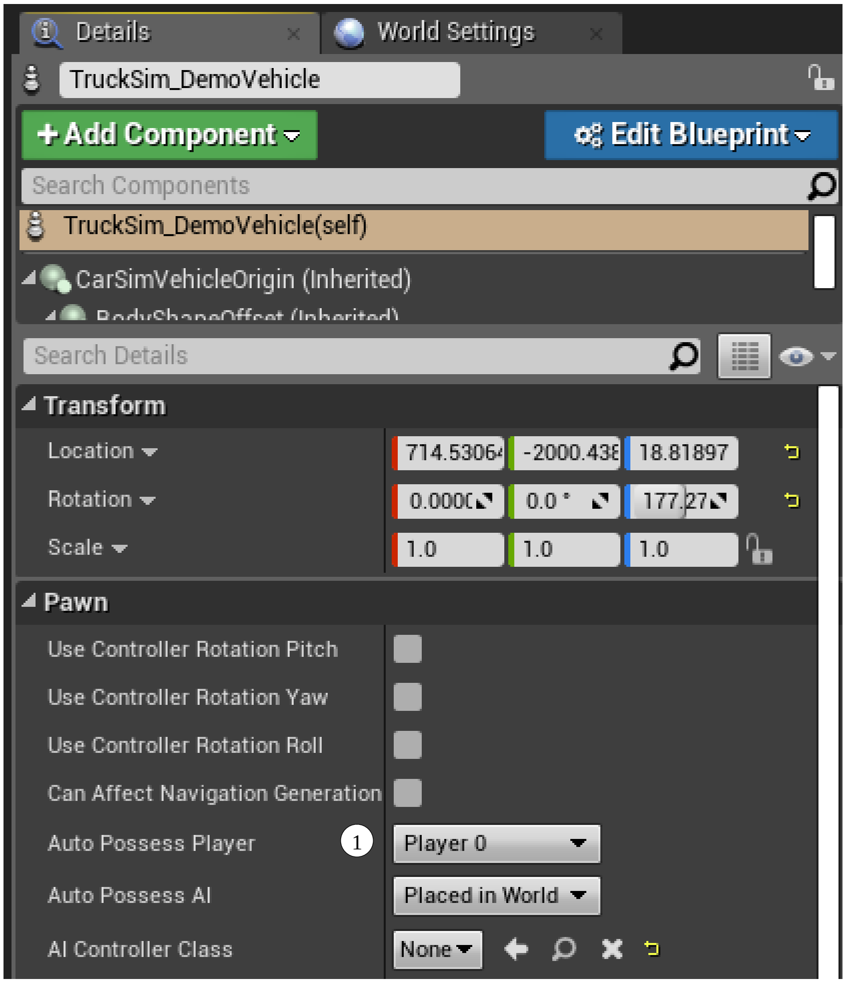
Figure 22. Changing the friction coefficient.

Using TruckSim in the plugin

New for the 2019.0 release, you can use TruckSim solvers with the plugin. The setup and configuration are the same as outlined above for CarSim, although the blueprints are different to accommodate trailer spawning and uses appropriate shapes for the default truck model. As with the car, you can customize these to your needs or create your own blueprints.

The example project is by default setup to run CarSim. The **Disable VehicleSim vehicle** checkbox (see Figure 14) is selected for the truck model, which will not allow the vehicle to initialize. The truck blueprint is configured to hide it from the scene when disabled.

If you want to use the truck you will need to turn off the **Disable VehicleSim vehicle** option and set the pawn to **Auto Possess Player** ① with your active player (Figure 23), usually **Player 0**. You may also want to disable and hide the car if you’re using the truck.



TruckSim_DemoVehicle

TruckSim_DemoVehicle(self)

CarSimVehicleOrigin (Inherited)

BodyShapeOffset (Inherited)

Transform

Location 714.5306 -2000.43 18.81897

Rotation 0.0000 0.0 ° 177.27

Scale 1.0 1.0 1.0

Pawn

Use Controller Rotation Pitch □

Use Controller Rotation Yaw □

Use Controller Rotation Roll □

Can Affect Navigation Generation □

Auto Possess Player Player 0

Auto Possess AI Placed in World

AI Controller Class None

Figure 23. Setting the TruckSim pawn to be possessed by Player 0.