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

RT-Lab Target Systems

This memo describes how to use a VehicleSim (VS) product (BikeSim, CarSim, or TruckSim) with an RT-Lab target system. The instructions are the same if you are using BikeSim or TruckSim. Note that you need the RT license for RT-Lab in order to use RT-Lab with a VS product.

System Requirements

We support RT-Lab 10.4.x and newer based on documented support from Opal-RT; we have tested RT-Lab 11.3 on Linux 32-bit and 2021.3.4 Linux 32/64-bit. Please note that we are not able to support breaking changes between versions that are produced by Opal-RT.

CarSim	TruckSim	BikeSim
2.0 GHz Dual Core	2.4 GHz Dual Core	2.0 GHz Dual Core

Note Any firewall should be configured to allow the host and real time target to communicate.

VehicleSim RT-Lab Product Installation

Working with RT-Lab requires VS Solvers for the vehicle math models compiled for the RT system, along with some datasets that connect to those RT VS Solvers. Before installing the RT-Lab part of the VS software, you must have the Windows version of the VS product installed. For example, you would first install CarSim for Windows, then install CarSim RT for RT-Lab on top of the Windows installation.

The installation of the VS RT extensions for RT-Lab includes program files compiled for the RT-Lab target and example datasets for the VS product.

Note Before starting CarSim for RT-Lab, you must have MATLAB Coder (formally Real-Time Workshop) and the RT-Lab software installed. Be sure you have the proper versions of software running for what you intend to accomplish.

RT-Lab does not allow the pathname of the Simulink model to contain any spaces. We recommend that your CarSim database be installed in a location where the pathname has no spaces. This will allow example Simulink models to run with minimal modification.

VS Solver Installation

To install the solver on the real-time target, follow the instructions below:

1. Launch CarSim using any installed database. Make sure to place a check mark next to `CarSim Solver for Opal-RT` in the **License Settings** window.
2. Go to **Tools > Install RT Solvers to Target…** ① (Figure 1). This opens the **Install Vehicle Solvers to Target** window (Figure 2).

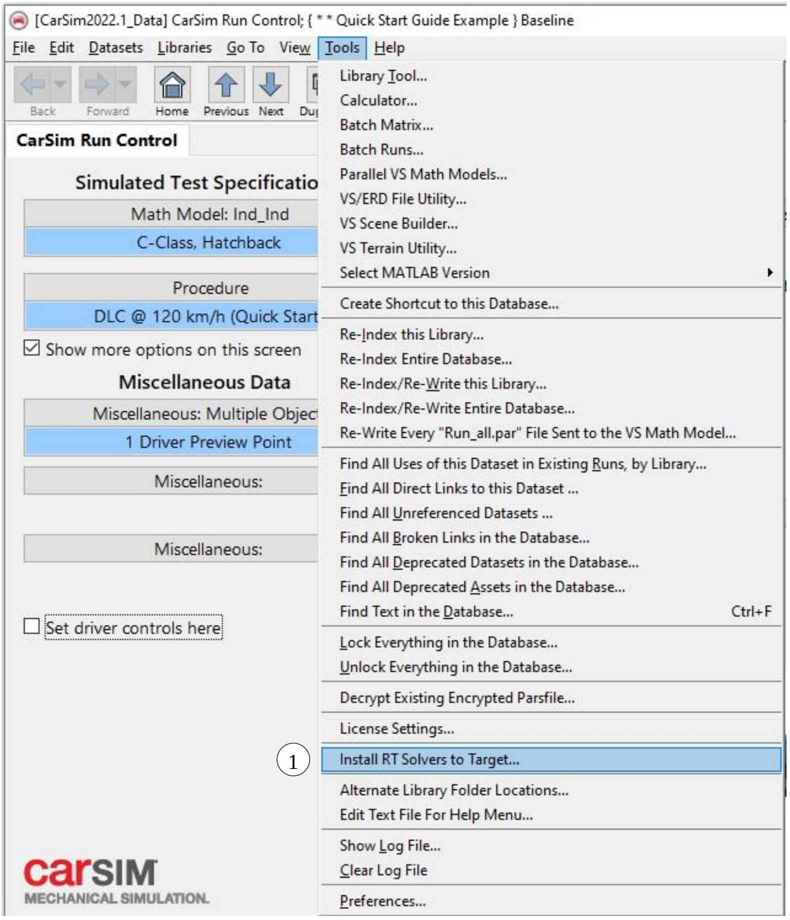
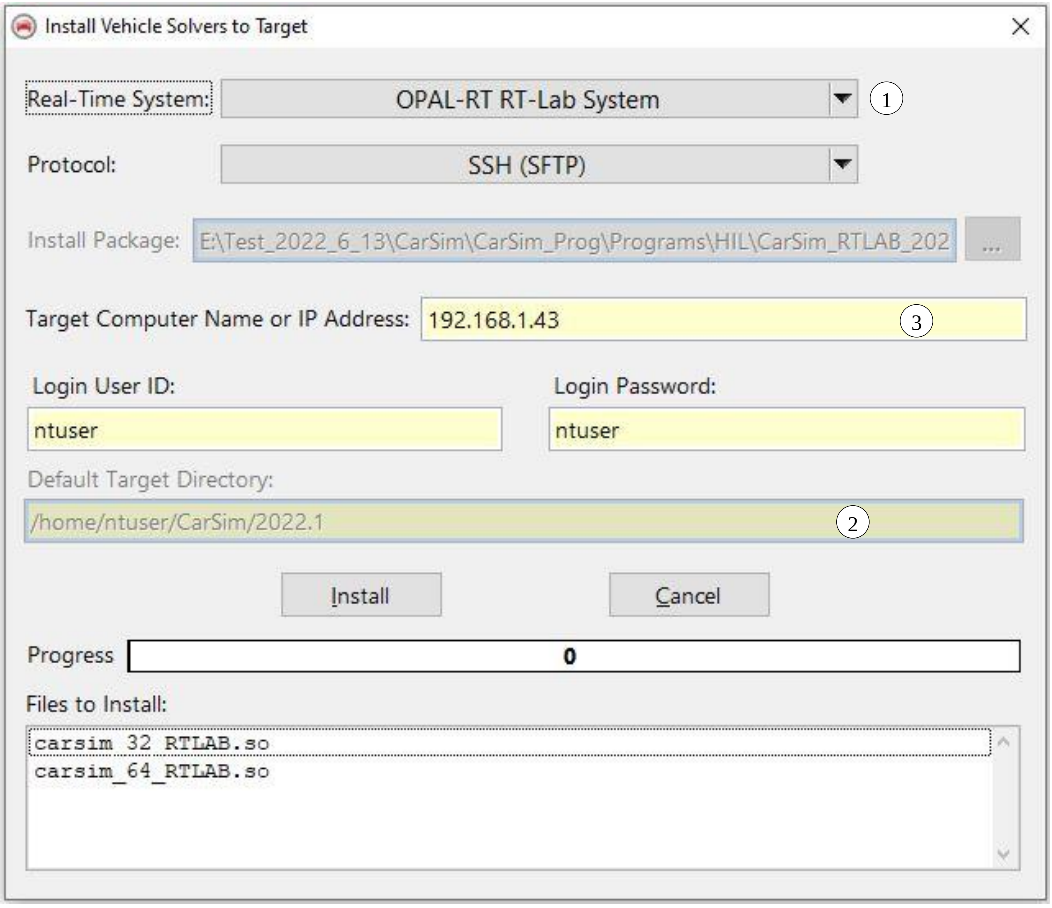

[CarSim2022.1_Data] CarSim Run Control; { * * Quick Start Guide Example } Baseline
File
Edit
Datasets
Libraries
Go To
View
Tools
Help
Back
Forward
Home
Previous
Next
Du
CarSim Run Control
Simulated Test Specificatio
Math Model: Ind_Ind
C-Class, Hatchback
Procedure
DLC @ 120 km/h (Quick Start
☑ Show more options on this screen
Miscellaneous Data
Miscellaneous: Multiple Objec
1 Driver Preview Point
Miscellaneous:
Miscellaneous:
☐ Set driver controls here
carSIM
MECHANICAL SIMULATION.
Library Tool...
Calculator...
Batch Matrix...
Batch Runs...
Parallel VS Math Models...
VS/ERD File Utility...
VS Scene Builder...
VS Terrain Utility...
Select MATLAB Version
Create Shortcut to this Database...
Re-Index this Library...
Re-Index Entire Database...
Re-Index/Re-Write this Library...
Re-Index/Re-Write Entire Database...
Re-Write Every "Run_all.par" File Sent to the VS Math Model...
Find All Uses of this Dataset in Existing Runs, by Library...
Find All Direct Links to this Dataset ...
Find All Unreferenced Datasets ...
Find All Broken Links in the Database...
Find All Deprecated Datasets in the Database...
Find All Deprecated Assets in the Database...
Find Text in the Database... Ctrl+F
Lock Everything in the Database...
Unlock Everything in the Database...
Decrypt Existing Encrypted Parsfile...
License Settings...
1
Install RT Solvers to Target...
Alternate Library Folder Locations...
Edit Text File For Help Menu...
Show Log File...
Clear Log File
Preferences...

Figure 1. Tools menu to install VS RT Solvers.



Install Vehicle Solvers to Target

Real-Time System: OPAL-RT RT-Lab System ①

Protocol: SSH (SFTP)

Install Package: E:\Test_2022_6_13\CarSim\CarSim_Prog\Programs\HIL\CarSim_RTLAB_202

Target Computer Name or IP Address: 192.168.1.43 ③

Login User ID: ntuser

Login Password: ntuser

Default Target Directory: /home/ntuser/CarSim/2022.1 ②

Progress 0

Files to Install:

```
carsim_32_RTLAB.so
carsim_64_RTLAB.so
```

Figure 2. Select a target RT System to install the Vehicle Solvers.

3. Select Opal-RT RT-Lab System from the Real-Time System pull-down menu ① (Figure 2). The solver installation package location is automatically detected ②. Confirm that the target computer, user ID, and password are correct. Click **Install**.

RT-Lab Example Datasets Installation

> **Note** In versions earlier than 2019.0, BikeSim Windows version contained the example datasets. Starting with version 2019.0, the example datasets are contained in a CPAR file that is included in the `Resources` folder. Follow the instructions in this section for the installation.

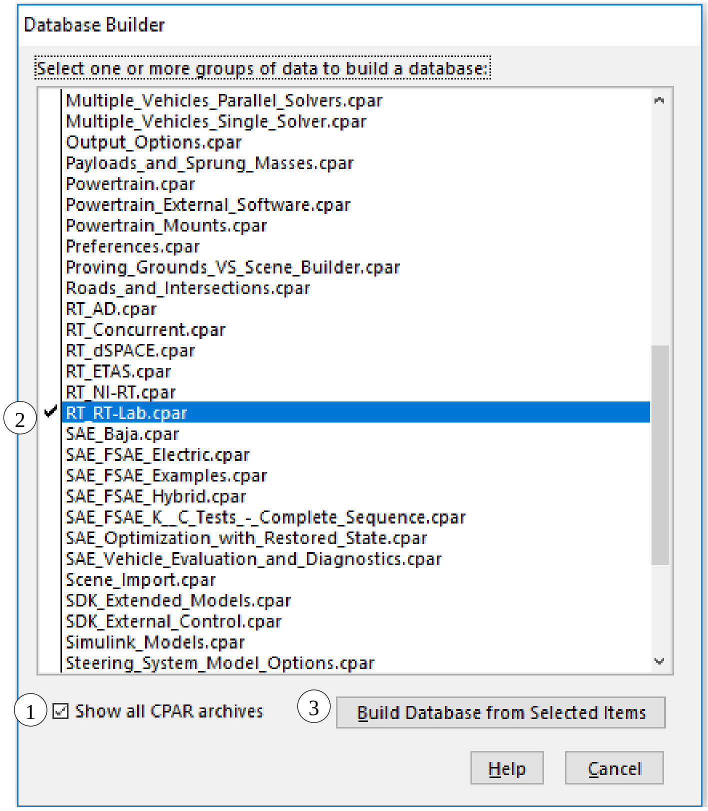
To install the VS Math Models and example datasets, you can use the Database Builder to build a new database, or you can import a consolidated parsfile (CPAR) to an existing database. Either method will install the necessary VS Math Models and supporting datasets.

To build a new database, launch CarSim and select **Open Database Builder…** from the Select Recent Database window. In the Database Builder window (Figure 3), toggle on **Show all CPAR archives** ①. Select RT_RT-Lab.cpar ② along with any other data groups that you would like to include in the new database. Click **Build Database from Selected Items**③. Select the folder where you would like to save the database and click **OK**.

To import a CPAR file to an existing database, launch CarSim and select the database that you will use with RT-Lab. In CarSim, navigate to **File** > **Import Parsfile (Any Export Type)** and import the following CPAR file:

```
CarSim_Prog\Resources\CPAR_Archives\RT_RT-Lab.cpar
```

Use the default settings (Figure 4) to import the file. The VS browser will copy files into the current database, import some dataset files, and refresh.



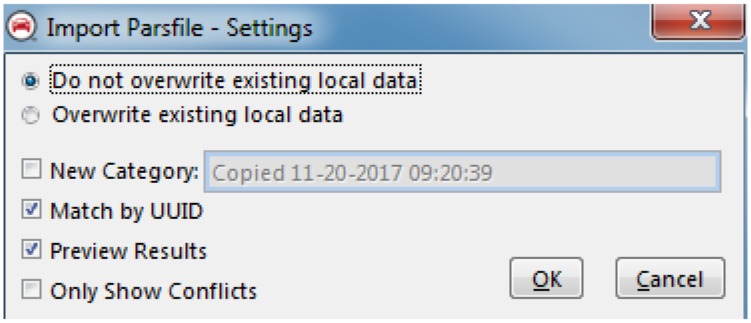
Database Builder

Select one or more groups of data to build a database:

Multiple_Vehicles_Parallel_Solvers.cpar
Multiple_Vehicles_Single_Solver.cpar
Output_Options.cpar
Payloads_and_Sprung_Masses.cpar
Powertrain.cpar
Powertrain_External_Software.cpar
Powertrain_Mounts.cpar
Preferences.cpar
Proving_Grounds_VS_Scene_Builder.cpar
Roads_and_Intersections.cpar
RT_AD.cpar
RT_Concurrent.cpar
RT_dSPACE.cpar
RT_ETAS.cpar
RT_NI-RT.cpar
✓ RT_RT-Lab.cpar
SAE_Baja.cpar
SAE_FSAE_Electric.cpar
SAE_FSAE_Examples.cpar
SAE_FSAE_Hybrid.cpar
SAE_FSAE_K__C_Tests_-_Complete_Sequence.cpar
SAE_Optimization_with_Restored_State.cpar
SAE_Vehicle_Evaluation_and_Diagnostics.cpar
Scene_Import.cpar
SDK_Extended_Models.cpar
SDK_External_Control.cpar
Simulink_Models.cpar
Steering_System_Model_Options.cpar

☑ Show all CPAR archives

Figure 3. Database Builder.



Import Parsfile - Settings

● Do not overwrite existing local data
○ Overwrite existing local data

□ New Category: Copied 11-20-2017 09:20:39
☑ Match by UUID
☑ Preview Results
□ Only Show Conflicts

Figure 4. Import Parsfile Default Settings.

Your dataset now includes examples for the RT-Lab system. The examples are found in **Datasets** > **RT: RT-Lab** (Figure 5).

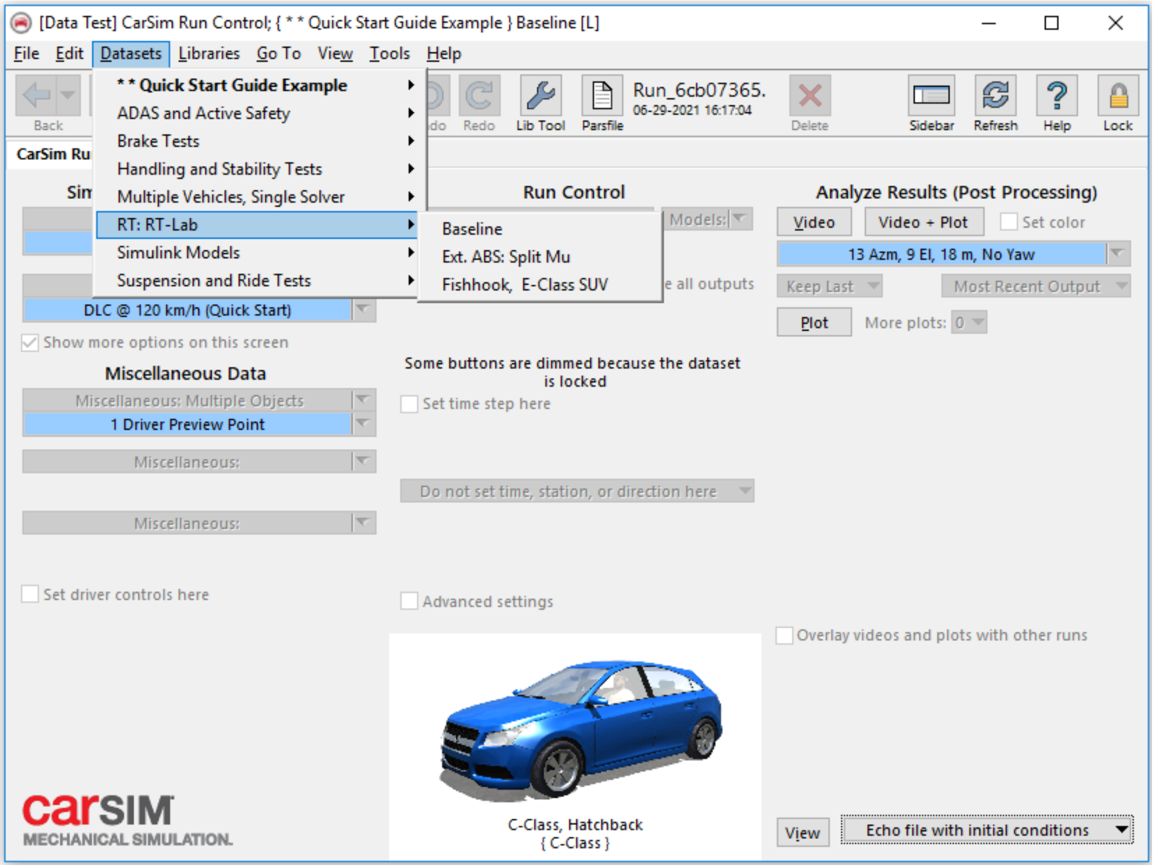
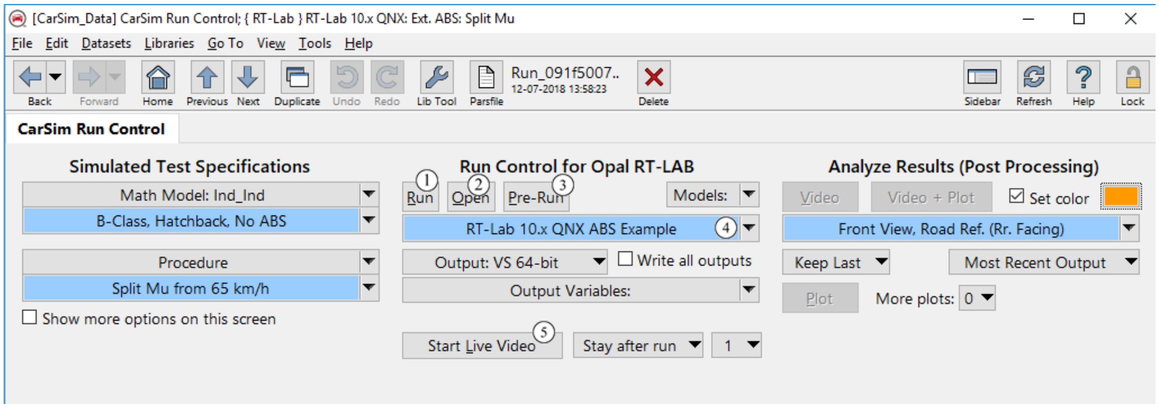

[Data Test] CarSim Run Control; { * * Quick Start Guide Example } Baseline [L]
File Edit Datasets Libraries Go To View Tools Help
* * Quick Start Guide Example
ADAS and Active Safety
Brake Tests
Handling and Stability Tests
Multiple Vehicles, Single Solver
RT: RT-Lab
Simulink Models
Suspension and Ride Tests
Baseline
Ext. ABS: Split Mu
Fishhook, E-Class SUV
Back
Redo
Lib Tool
Parsfile
Run_6cb07365.
06-29-2021 16:17:04
Delete
Sidebar
Refresh
Help
Lock
Run Control
Analyze Results (Post Processing)
Models:
Video
Video + Plot
Set color
13 Azm, 9 El, 18 m, No Yaw
Keep Last
Most Recent Output
Plot
More plots: 0
DLC @ 120 km/h (Quick Start)
Show more options on this screen
Miscellaneous Data
Miscellaneous: Multiple Objects
1 Driver Preview Point
Miscellaneous:
Miscellaneous:
Set driver controls here
Some buttons are dimmed because the dataset is locked
Set time step here
Do not set time, station, or direction here
Advanced settings
Overlay videos and plots with other runs
C-Class, Hatchback
{ C-Class }
View
Echo file with initial conditions
carSIM
MECHANICAL SIMULATION.

Figure 5. RT-Lab Examples.

Making Runs With RT-Lab Version 10.x

Run Control for RT-Lab in CarSim GUI

The CarSim Run Control screen for RT-Lab is shown here (Figure 6). Descriptions of the most used buttons pertaining to running RT-Lab with CarSim are listed below.



Math Model: Ind_Ind

B-Class, Hatchback, No ABS

Procedure

Split Mu from 65 km/h

☐ Show more options on this screen

RT-Lab 10.x QNX ABS Example

Output: VS 64-bit

☐ Write all outputs

☑ Set color

Front View, Road Ref. (Rr. Facing)

Figure 6. CarSim RT-Lab Run Control with ABS controller

① **Run**: This will generate the latest Parsfile(s), start the live animation, and start the "Run RT-Lab 10.x Target" program, which will load the model and Parsfile(s) to the target and initialize the model.

② **Open**: This will open the Simulink model associated with the current dataset. This will also generate the parsfile(s) and create the .llm file, which allows the corresponding RT-Lab project to automatically set the correct library and target to match the CarSim Settings.

③ **Pre-Run**: This will generate the most current parsfile(s).

④ **DataSet Link**: This links to the RT-Lab Models dataset screen (Figure 8).

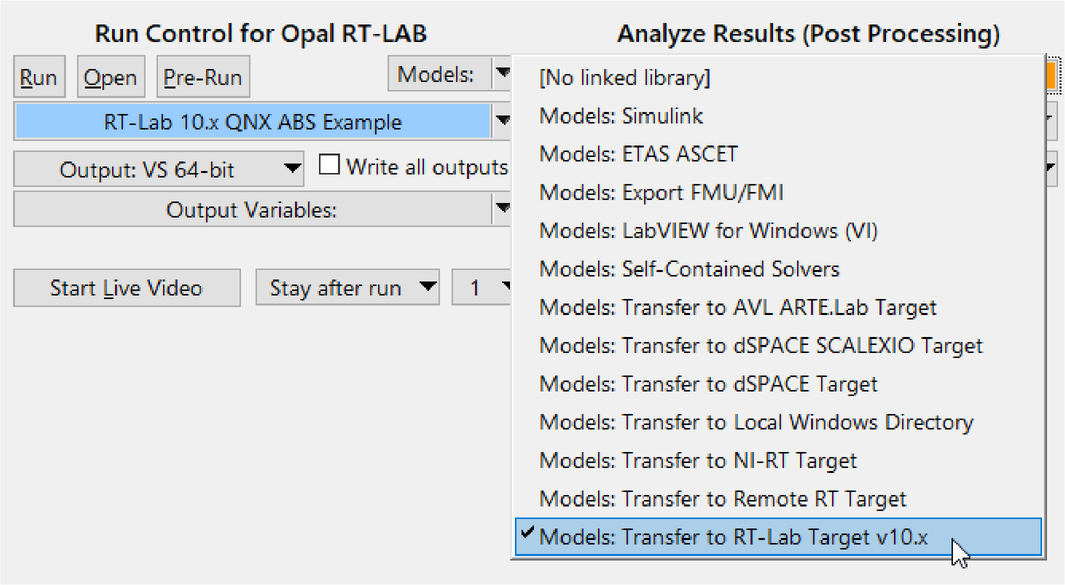
⑤ **Start Live Video**: This will open the live animation currently set in CarSim VS_Visualizer.

Set Up Model with RT-Lab Hardware Environments

Before you start to build and run any examples, you must configure your hardware environments by providing IP addresses for the host and target machines, your hardware physical node name, and the location of your Simulink model file.

1. Start CarSim.
2. To run an example of ABS controller, on the CarSim **Run Control** screen, use the **Dataset** menu to go to the example **RT: RT Lab–> Ext. ABS: Split Mu** (Figure 6).

Note If you want to create a new dataset rather than use a pre-existing example, instead select **Models: Transfer to RT-Lab Target v10.x** (Figure 7). You can use the **Dataset** menu to **[Link to New Dataset]** and create your new dataset.


Run Control for Opal RT-LAB
Analyze Results (Post Processing)
Run
Open
Pre-Run
Start Live Video
Stay after run
1
Output Variables:
[No linked library]
Models: Simulink
Models: ETAS ASCET
Models: Export FMU/FMI
Models: LabVIEW for Windows (VI)
Models: Self-Contained Solvers
Models: Transfer to AVL ARTE.Lab Target
Models: Transfer to dSPACE SCALEXIO Target
Models: Transfer to dSPACE Target
Models: Transfer to Local Windows Directory
Models: Transfer to NI-RT Target
Models: Transfer to Remote RT Target

Models: RT-Lab 10.x QNX ABS Example

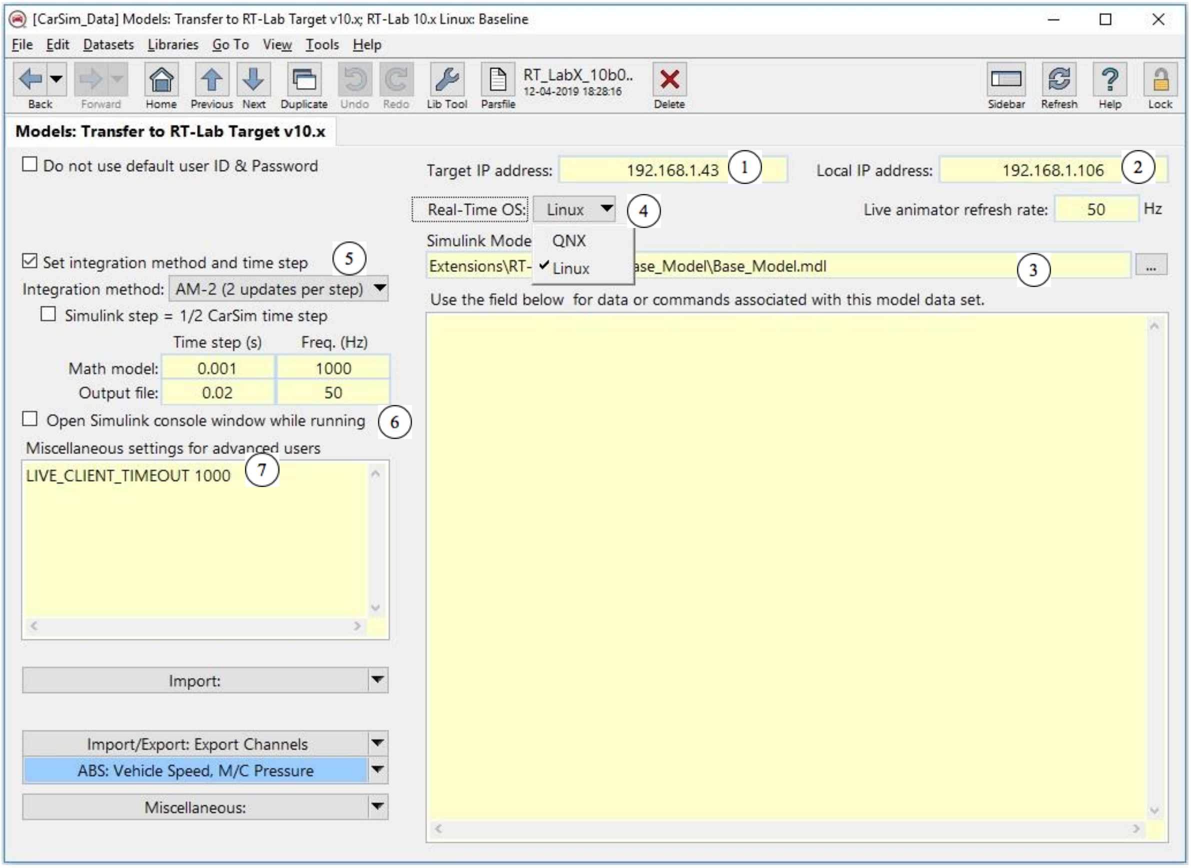
Output: VS 64-bit

□ Write all outputs

✓ Models: Transfer to RT-Lab Target v10.x

Figure 7. Example select model Target.

3. Click on the Models blue link for the ABS example to view the linked dataset (Figure 8).



[CarSim_Data] Models: Transfer to RT-Lab Target v10.x; RT-Lab 10.x Linux: Baseline

RT_LabX_10b0..
12-04-2019 18:28:16

Models: Transfer to RT-Lab Target v10.x

□ Do not use default user ID & Password

Target IP address: 192.168.1.43

Local IP address: 192.168.1.106

Real-Time OS: Linux

Live animator refresh rate: 50 Hz

Simulink Mode

Extensions\RT- ✓ Linux ase_Model\Base_Model.mdl

☑ Set integration method and time step

Integration method: AM-2 (2 updates per step)

□ Simulink step = 1/2 CarSim time step

	Time step (s)	Freq. (Hz)
Math model:	0.001	1000
Output file:	0.02	50

□ Open Simulink console window while running

Miscellaneous settings for advanced users

```
LIVE_CLIENT_TIMEOUT 1000
```

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

Import/Export: Export Channels

ABS: Vehicle Speed, M/C Pressure

Figure 8. Dataset for RT-Lab example with ABS controller.

4. Set the IP addresses for the target ① and host ②.
5. Set the location of the Simulink model ③.
6. Set the **Real-Time OS** ④. If you are using a QNX OS on your target machine, select QNX. If you are using a Redhat Linux OS, select Linux.
7. Set the integration method to AM-2 with Math model time step of 0.001⑤.
8. After the model has been loaded in RT-Lab as described later in this document, the Live Animation connection to the VS Visualizer will only stay active for a short time. If you press “Execute” in RT-Lab after the VS Visualizer has disconnected, RT-Lab will reset. To extend the time that VS Visualizer will stay connected to RT-Lab, you can enter ‘live_client_timeout [*time in seconds]*’ in the yellow field ⑦. For example, ‘live_client_timeout 1000’ tells the VS Visualizer to wait 1000 seconds before disconnecting.
9. Return to the **Run Control** screen.

Note SurfAnim is no longer provided or supported by Mechanical Simulation. Therefore, for Redhat Linux targets, change the CarSim live animation settings from `Tools >> Preferences` to use VsVisualizer.

Example: Build the Simulink model

Edit and verify your Simulink model by clicking the **Open** button (Figure 6) to open the Simulink model identified in the linked Model dataset (see the Simulink pathname ③ in Figure 8). The model should open in Simulink (Figure 9).

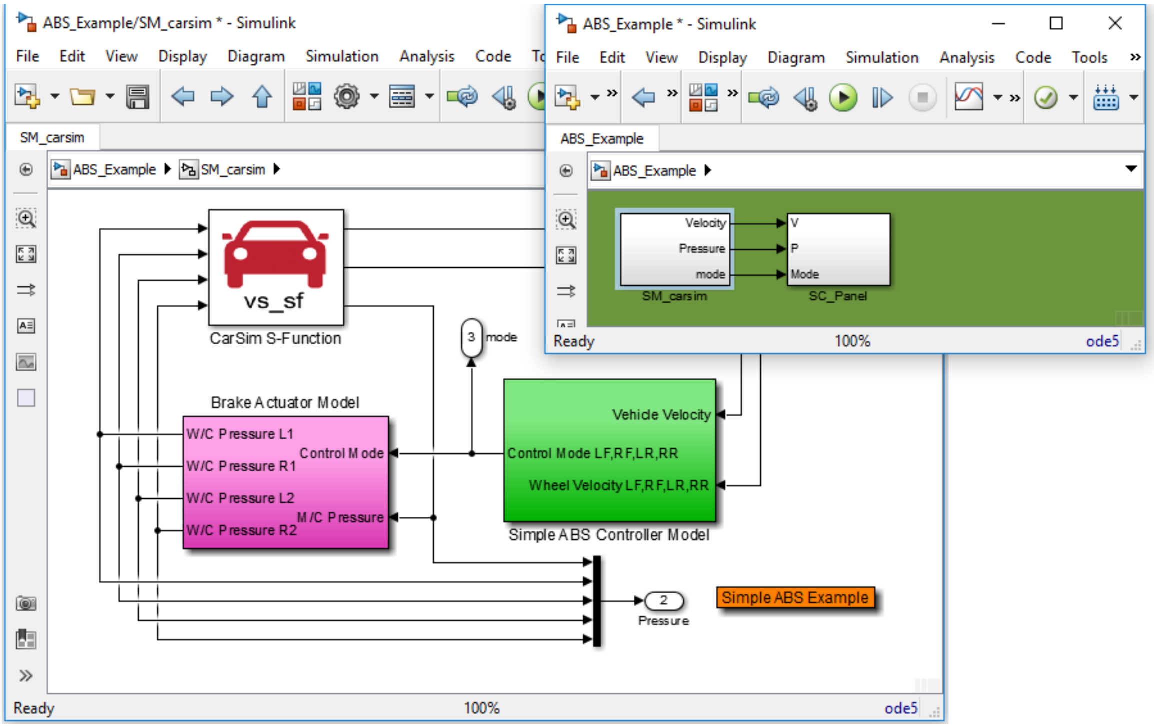


Figure 9. Simulink model for ABS example.

Opening the model in Simulink will also generate the parsfile(s) and create the .llm file that will allow RT-Lab to automatically set the correct settings to match the settings made in CarSim.

Click the run button (▶) on the Simulink toolbar (or type **Ctrl+T**) to confirm that the model settings are correct.

If the run is successful, you can now proceed to import the model into the "RT-Lab GUI" software to build and run.

Example: Build on the RT-Lab Target

After the "RT-Lab GUI" software is opened, you must create a new project and add the Simulink model selected earlier in CarSim. In the "RT-Lab GUI" software, go to **File->New->RT-Lab Project**, input a project name, click **Next**, and choose **Empty** from the list of existing templates.

The new project that you make should show up in the RT-Lab **Project Explorer**. Expand the project folder, right click **Models**, and then go to **Add–>Existing model** (Figure 10).

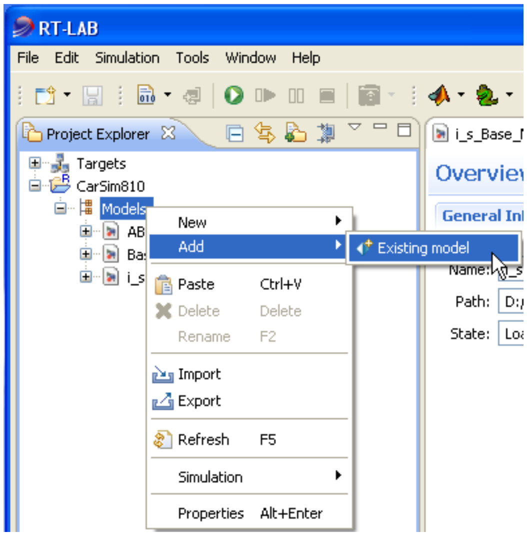
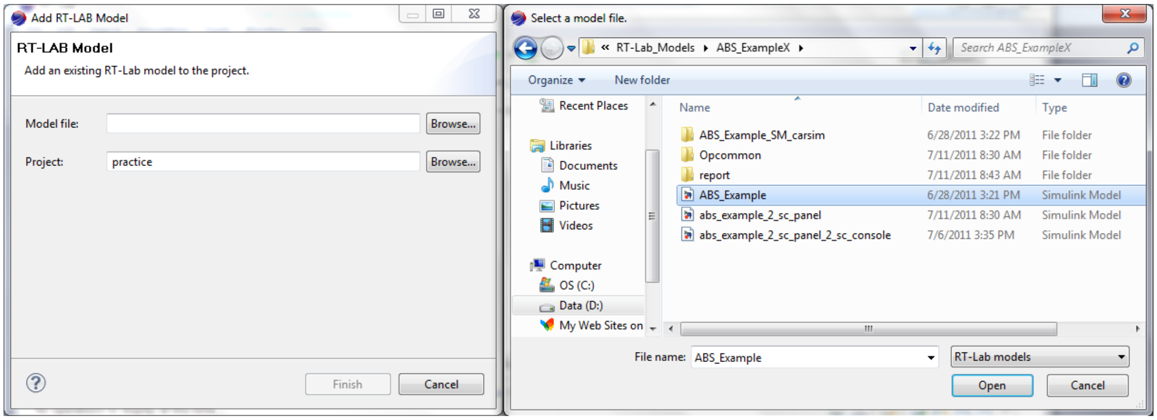


Figure 10. Adding existing model in "RT-Lab GUI" software.

This will open the **Add RT-LAB Model** window. Hit the **Browse...** button next to the **Model** file section to open the explorer, where you can search for the Simulink x.mdl file you want. In this example, it is `CarSim_Data\Extensions\RT-Lab_Models\ABS_Example\ABS_Example` (Figure 11).



Model file:

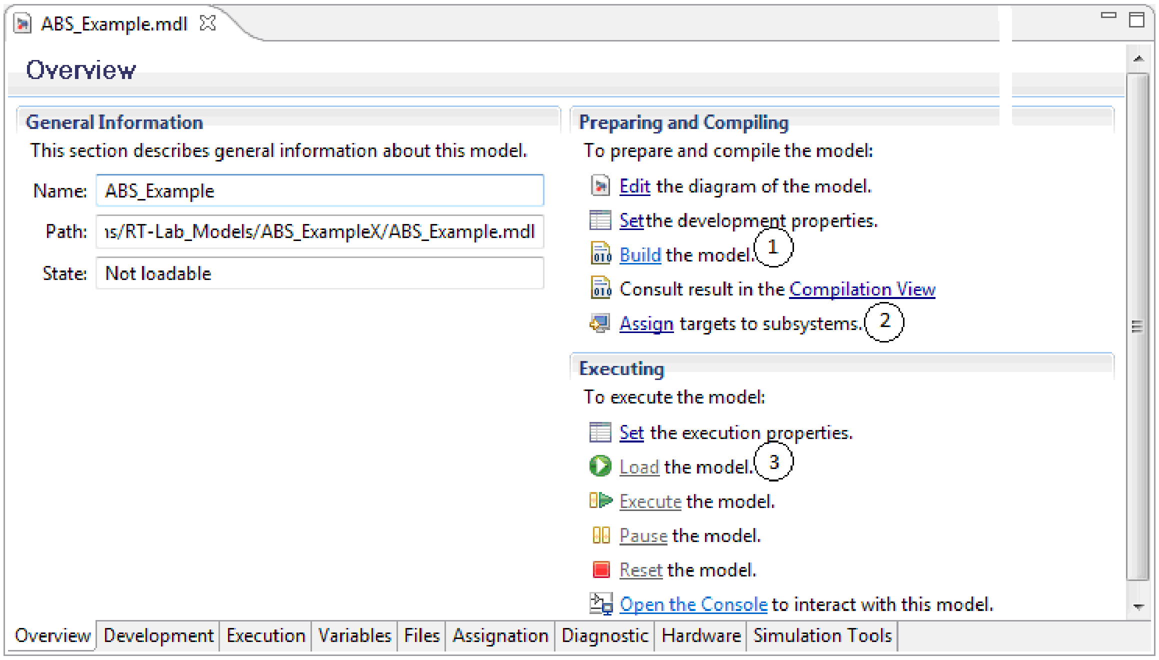
Project: practice

Name	Date modified	Type
ABS_Example_SM_carsim	6/28/2011 3:22 PM	File folder
Opcommon	7/11/2011 8:30 AM	File folder
report	7/11/2011 8:43 AM	File folder
ABS_Example	6/28/2011 3:21 PM	Simulink Model
abs_example_2_sc_panel	7/11/2011 8:30 AM	Simulink Model
abs_example_2_sc_panel_2_sc_console	7/6/2011 3:35 PM	Simulink Model

File name: ABS_Example

Figure 11. Adding ABS-Example Model file.

This will open **ABS_Example.mdl** in the "RT-Lab GUI" software (Figure 12).



General Information

This section describes general information about this model.

Name: ABS_Example

Path: ıs/RT-Lab_Models/ABS_ExampleX/ABS_Example.mdl

State: Not loadable

Preparing and Compiling

To prepare and compile the model:

- Edit the diagram of the model.
- Setthe development properties.
- Build the model.
- Consult result in the Compilation View
- Assign targets to subsystems.

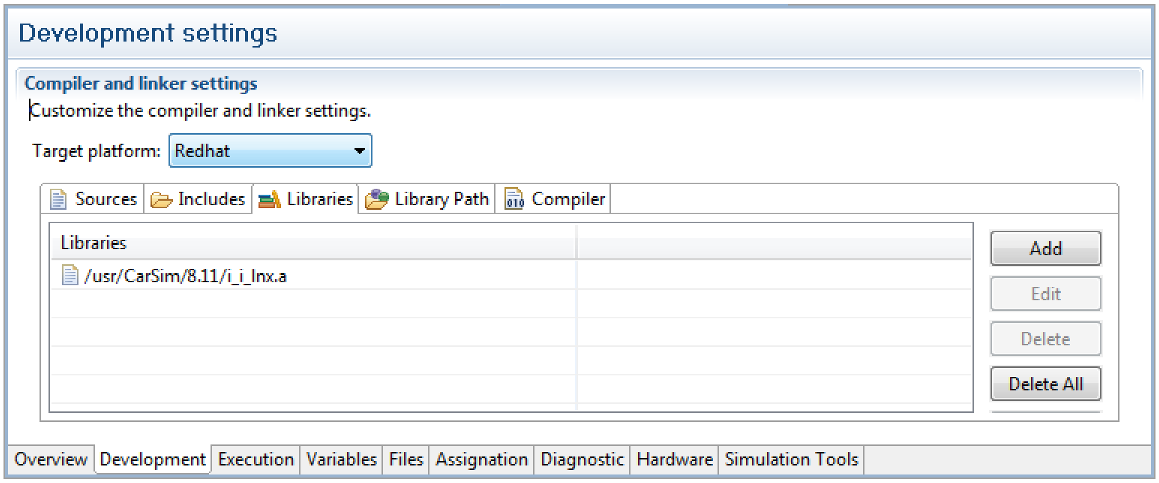
Executing

To execute the model:

- Set the execution properties.
- Load the model.
- Execute the model.
- Pause the model.
- Reset the model.
- Open the Console to interact with this model.

Figure 12. Model Overview in "RT-Lab GUI" software.

Go to the **Development** tab (Figure 13). Here, make sure the **Target Platform** is in fact the one you set in CarSim (Figure 8). Then click on **Libraries**. The name of the listed library should match what you have set in CarSim. In this example, the vehicle configuration set in CarSim is i_i and the target machine is a Redhat Linux platform, so the library's name of i_i_lnx.a does match the CarSim setting.



Compiler and linker settings

Customize the compiler and linker settings.

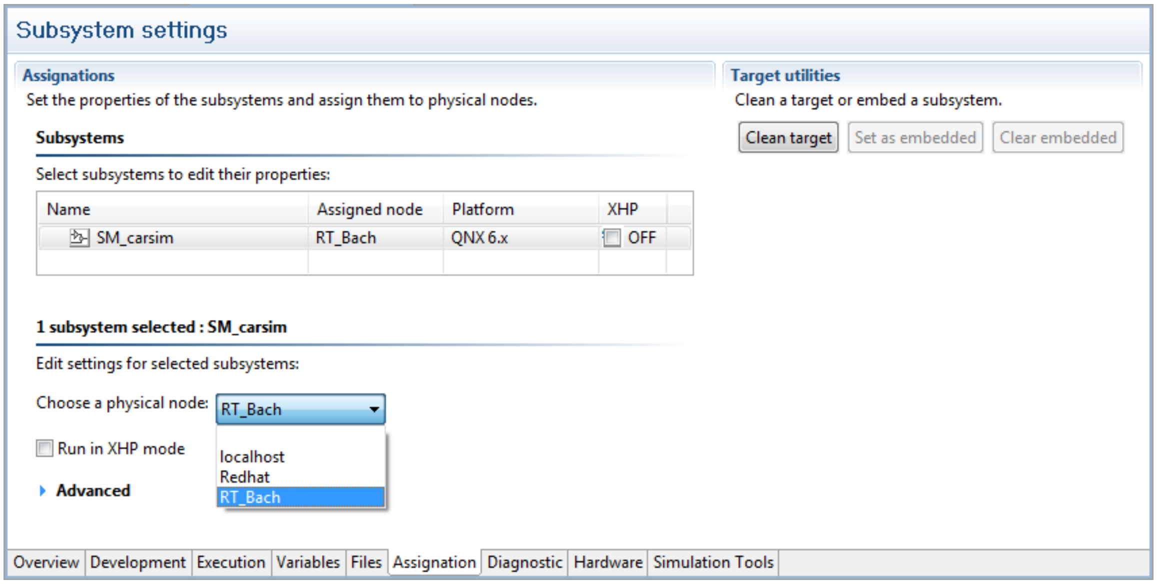
Target platform: Redhat

Libraries
/usr/CarSim/8.11/i_i_lnx.a

Figure 13. RT-Lab Development Settings: Libraries

Go back to the **Overview** tab (Figure 12) and click **Build** ①. When the model is done building, click **Assign targets to subsystems** ②.

On the **Assignation** tab (Figure 14), click on **SM_carsim** under **Subsystems**. Then, make sure the assigned node matches the target system you have (In this example, **RT_bach** for QNX, **Redhat** for Redhat Linux).



Subsystem settings

Assignations

Set the properties of the subsystems and assign them to physical nodes.

Subsystems

Select subsystems to edit their properties:

Name	Assigned node	Platform	XHP
SM_carsim	RT_Bach	QNX 6.x	□ OFF

1 subsystem selected : SM_carsim

Edit settings for selected subsystems:

Choose a physical node: RT_Bach

□ Run in XHP mode

Advanced

Target utilities

Clean a target or embed a subsystem.

Figure 14. RT-Lab node assignation.

Now, before we continue and load the model, we should go back to CarSim. Look at the **Run Control** area and check the number of live animators for this run. If the number is 1 or more, make sure to click the **Start Live Video** button to open the "Live Animation" program.

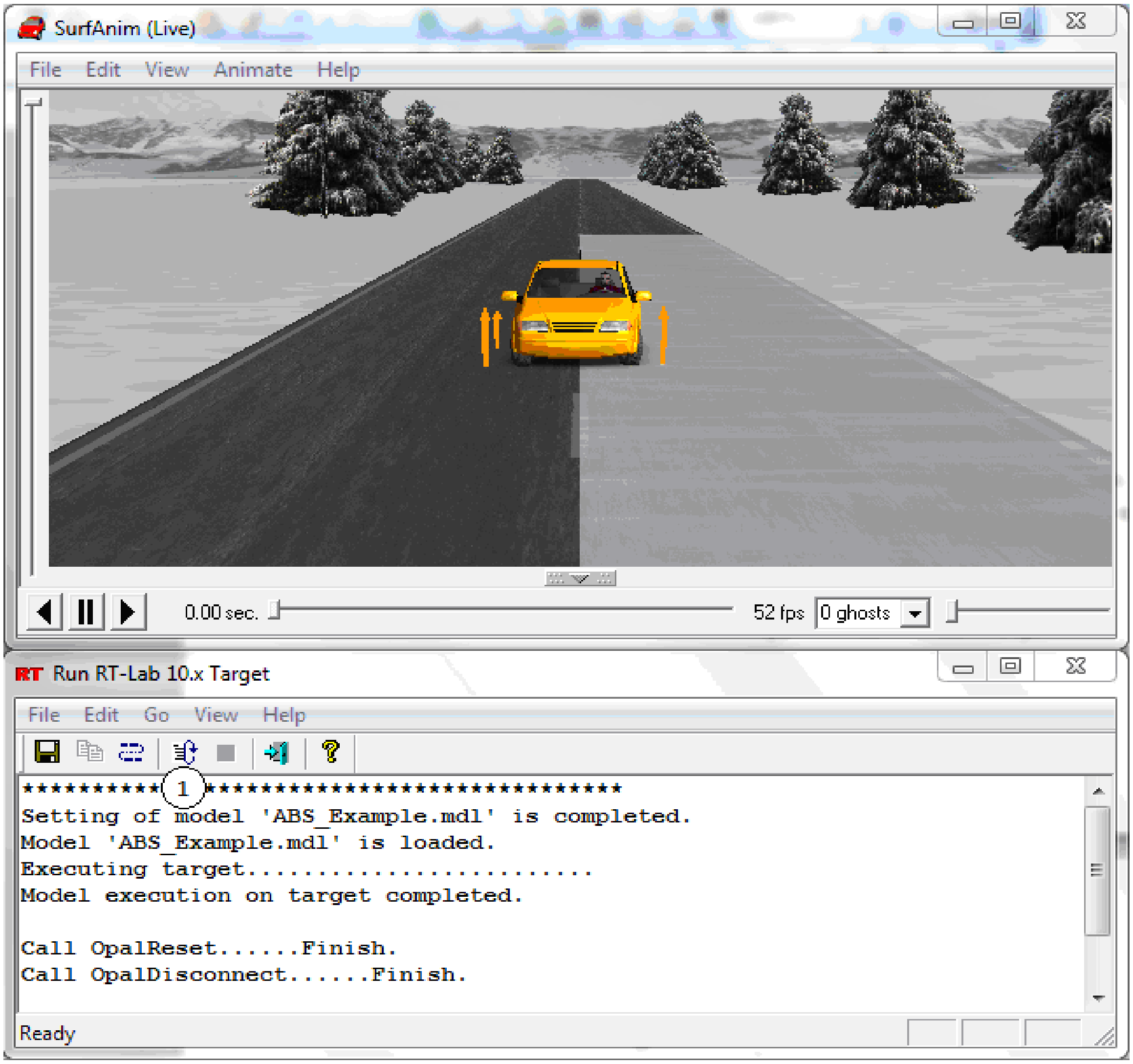
Now, go back to the **Overview** tab in the "RT-Lab GUI" software and click **Load** ③. This will load the model and Parsfile to the target and initialize the model, after which you can click **Execute** to start the animation (Figure 15).

Note If you do not click on **Execute** soon after the model has loaded, the VS Visualizer will disconnect from RT-LAB. Clicking on **Execute** after the Visualizer has disconnected will cause RT-LAB to reset. To prevent the VS Visualizer from disconnecting and RT-LAB from resetting, enter 'live_client_timeout [*time in seconds]*'in the yellow field on the VehicleSim Model screen as shown in Figure 8 ⑦ . The live_client_timout keyword sets the length of time in seconds that the VS Visualizer will wait before disconnecting. For example, 'live_client_timeout 100' causes the VS Visualizer to wait 100 seconds before disconnecting.

Run Pre-Built Model

When the model has already been built, you can make a new run through the methods listed below without changing the Simulink model or RT-Lab settings.

1. Go to CarSim and click the **Run** button on the **Run Control** screen (Figure 6). This will generate the latest Parsfile(s), start the live animation, and start the "Run RT-Lab 10.x Target" program, which will load the model and Parsfile(s) to the target and initialize the model.



```
**********  1  ******************************
Setting of model 'ABS_Example.mdl' is completed.
Model 'ABS_Example.mdl' is loaded.
Executing target.........................
Model execution on target completed.

Call OpalReset......Finish.
Call OpalDisconnect......Finish.
```

Figure 15. Run RT-Lab Target.

2. Click **Pre-Run** in CarSim to generate the latest Parsfile(s). Also make sure to open the "Live Animation" program. From here, you can use the "RT-Lab GUI" software to run the model by clicking **Load** and **Execute** or you can use the "RT-Lab 10.x Target" program (Figure 15) by clicking the rerun button ⟳ (1).

Optional Features

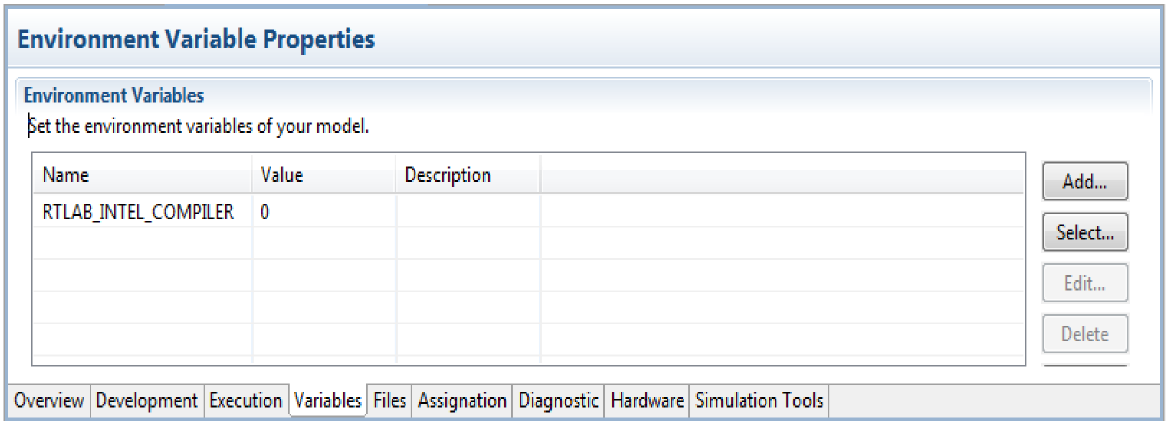
1. If your Simulink model has a console subsystem (SC_Panel in the ABS_Example window in Figure 9), you can choose whether or not to open it via the **Open Simulink console window while running** checkbox (6) in the CarSim Database window (Figure 8).

2. You can choose whether Matlab stays open or not after the simulation finishes. You can do this in the "Run RT-Lab 10.x Target" program (Figure 15) through **File–>Keep Matlab Open**. You can also set it in the "RT-Lab" GUI software through the **Simulation Tools** tab.

Using gcc compiler on Redhat

You can use the default gcc compiler that comes as a standard on the Redhat installation. To select the gcc compiler to be used, you need to set a specific user variable in RT-LAB, under the Variables tab of your model.

RTLAB_INTEL_COMPILER and set it to **0**



Environment Variable Properties

Environment Variables

Set the environment variables of your model.

Name	Value	Description
RTLAB_INTEL_COMPILER	0	

Note Removing the variable will bring back the Intel compiler as the compiler to be used.

You will not need to make any further changes. The CarSim libraries for Redhat Linux are cross compatible between the Intel compiler and gcc compiler.