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

Concurrent RT Guide for VehicleSim Products

Introduction

This memo describes how to use a Concurrent RT system with the real-time versions of BikeSim, CarSim, and TruckSim. The examples shown were made with CarSim RT, but the same controls and methods are also used for BikeSim RT and TruckSim RT.

To run a simulation using a VehicleSim model with Concurrent, you must have:

- MATLAB/Simulink/MATLAB Coder/Simulink Coder (formerly Real-Time Workshop) or use FMU/FMI 2.0.
- Concurrent software SIMulation Workbench, SimWB.

This Technical Memo assumes you have some familiarity with Concurrent software SimWB, Matlab/Simulink, FMU, and the desktop version of your VehicleSim product. If you are a new

VehicleSim user, we recommend that you begin with the Quick Start Guide for your product, and the *Running a VS Math Model in Simulink* tutorial. Both documents are available from the **Help** menu of your VehicleSim product.

System Requirements

We support Concurrent Redhawk 32bit/64bit Real-Time system, Linux Real-Time system, with SIMulation Workbench (SimWB). For 32bit Linux from Redhawk 5.4 with SimWB 6.0 or newer. For 64bit Linux from Redhawk 6.3 with SimWB 7.2 or newer.

CarSim	TruckSim	BikeSim
2.4 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.

Initial Setup

Before installing VehicleSim products, you need to enable FTP and create a user account on the Concurrent Linux target system.

Enable FTP Connection

To enable an FTP connection, you must have access to a Linux target machine.

1. Log into the Linux target machine as `root`.
2. Go to **System** > **Administration** > **Services** and select **vsftpd**. Change this to **Enable**.

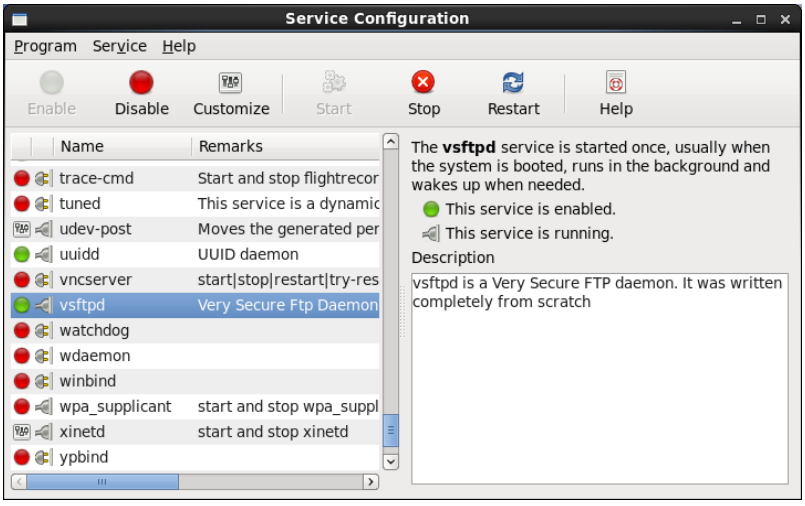

*Figure 1. Enable ftp service **vsftpd***

Create a User Account

You can create a Linux User account using the graphical *User Manager* application or by typing into a shell prompt in Linux. Here are the steps for using the shell prompt:

1. Open a terminal.
2. If you are not logged in as root, type the command `su` and the password of "`su`". The default password is

   `redhawk`
3. Now, we are going to create your user name with the following command:

   `useradd VehicleSim` (note: V and S are capital letters).
4. Now, create a password for the user "`VehicleSim`" with this command:

   `passwd VehicleSim`
5. This will prompt you to set a new password. Type: `vehiclesim`

> **Note** We have configured VehicleSim files under the assumption that you use `VehicleSim` as a user name. We strongly advise that you create an account with this exact name.

CarSim VS Solvers and Datasets Installation

In versions earlier than 2018.0, CarSim solvers and datasets for Concurrent were either downloaded from carsim.com or obtained from the installation CD. Starting with version 2018.0, all necessary files are packaged within **the CarSim for Windows** application.

VS Solver Installation

To install the solver, follow the instruction below:

1. Launch CarSim using any installed database (e.g., `CarSim2019.1_Data`). Make sure to place a check mark next to `CarSim solver for Linux-RT` in **License Setting Windows**.
2. Go to **Tools** > **Install RT Solvers to Target…** (Figure 2). This opens a solver installation window (Figure 3).

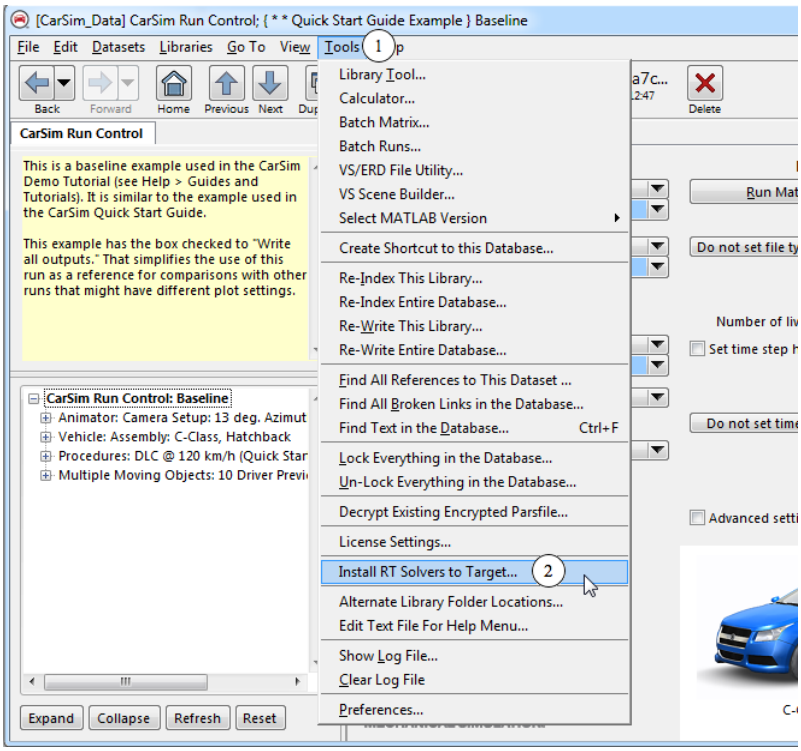

Figure 2. Tools menu to install VS RT solvers.

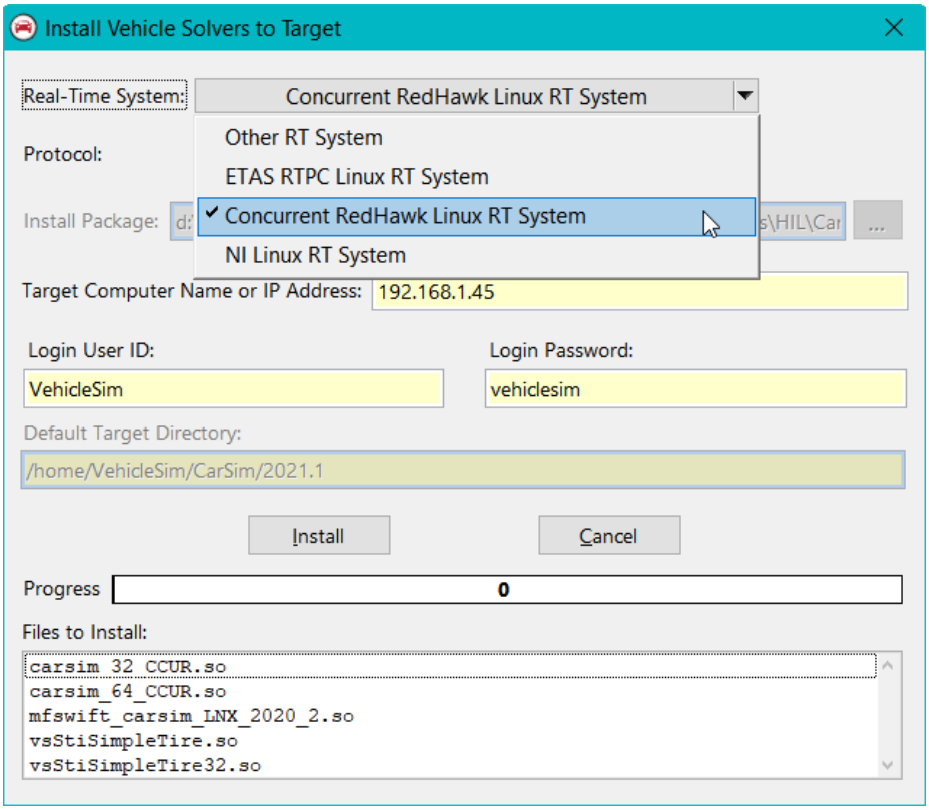

Figure 3. Select a target RT System to install Vehicle Solvers.

3. Select **Concurrent RedHawk Linux RT System** from the **Real-Time System** pull down menu. The solver installation package location is automatically detected (Figure 4). Make sure it is pointing to a correct Target address. The user ID and the password should be `VehicleSim` and `vehiclesim`, respectively, to match the username account you created in the previous section. Click **Install**.

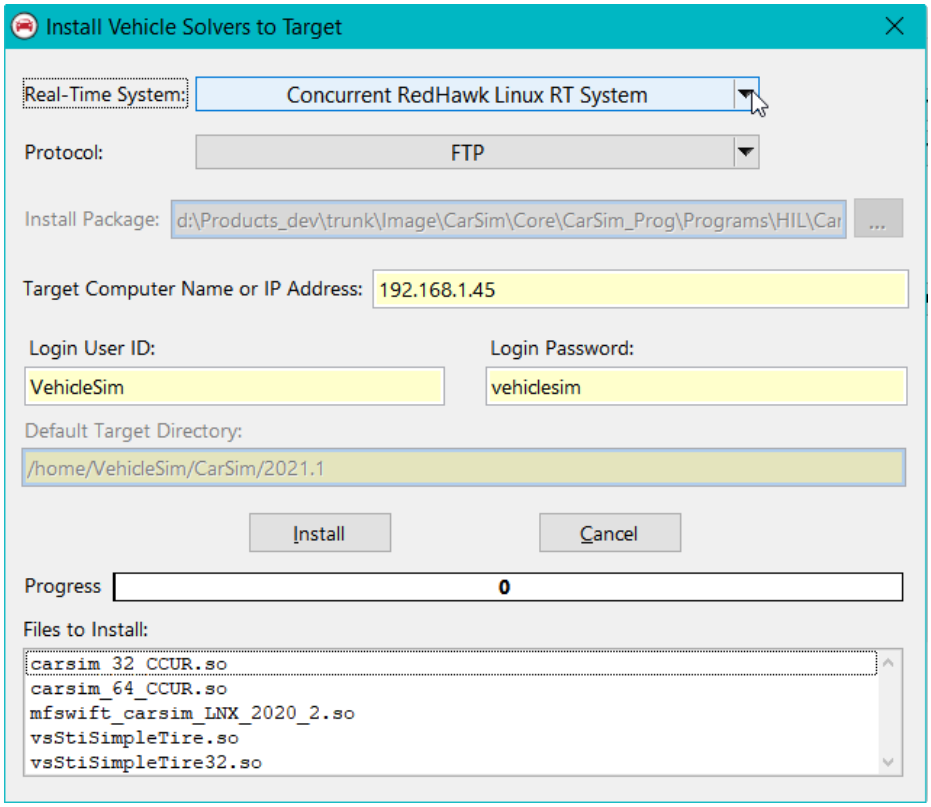

Figure 4. Install Vehicle Solvers to Concurrent RT system target.

Example Datasets Installation

The VS Math Models and example datasets for Concurrent can be installed by using the Database Builder to build a new database or by importing a consolidated parsfile (CPAR) to an existing database. Either method will install the necessary VS Math Models, corresponding Concurrent files, and supporting VehicleSim datasets.

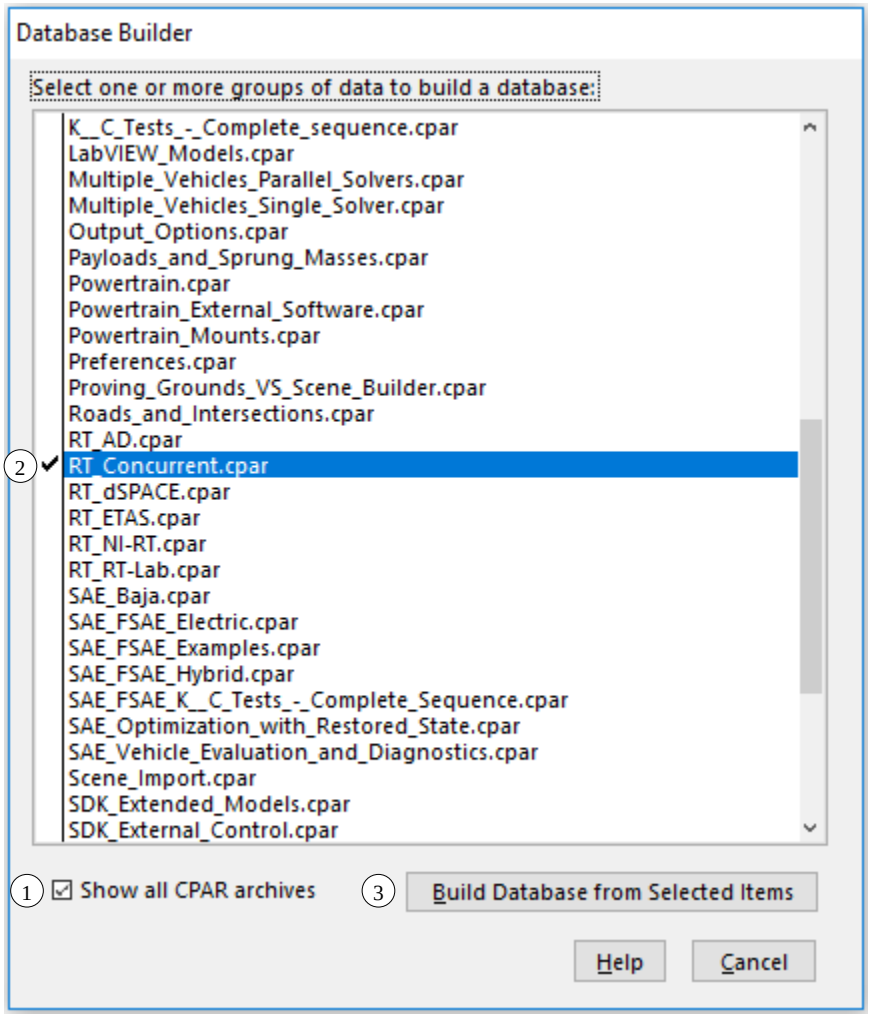

Figure 5. Database Builder

To build a new database, launch CarSim and select **Open Database Builder…** from the Select Recent Database window. In the Database Builder window (Figure 5), toggle on **Show all CPAR archives** (1). Select RT_Concurrent.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 Concurrent. In CarSim, navigate to **File** > **Import Parsfile (Any Export Type)** and import the following CPAR file:

`CarSim_Prog\Resources\CPAR_Archives\RT_Concurrent.cpar`

Use the default settings (Figure 6) to import the file. The VS browser will copy files into the current database, import some dataset files, and refresh.

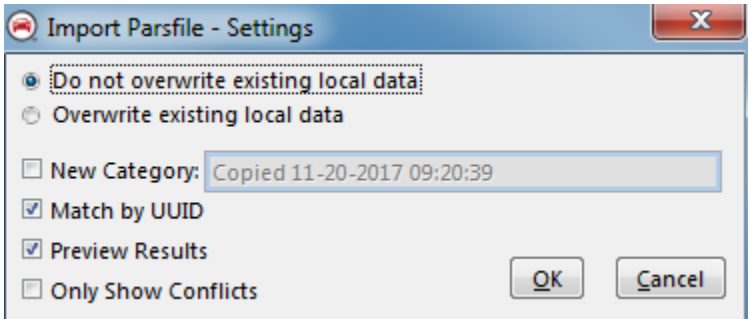

Figure 6. Import Parsfile Default Settings.

After building a new database or importing the consolidated parsfile, your dataset now includes examples for the Concurrent system. The examples are found in **Datasets > RT: Concurrent** (Figure 7).

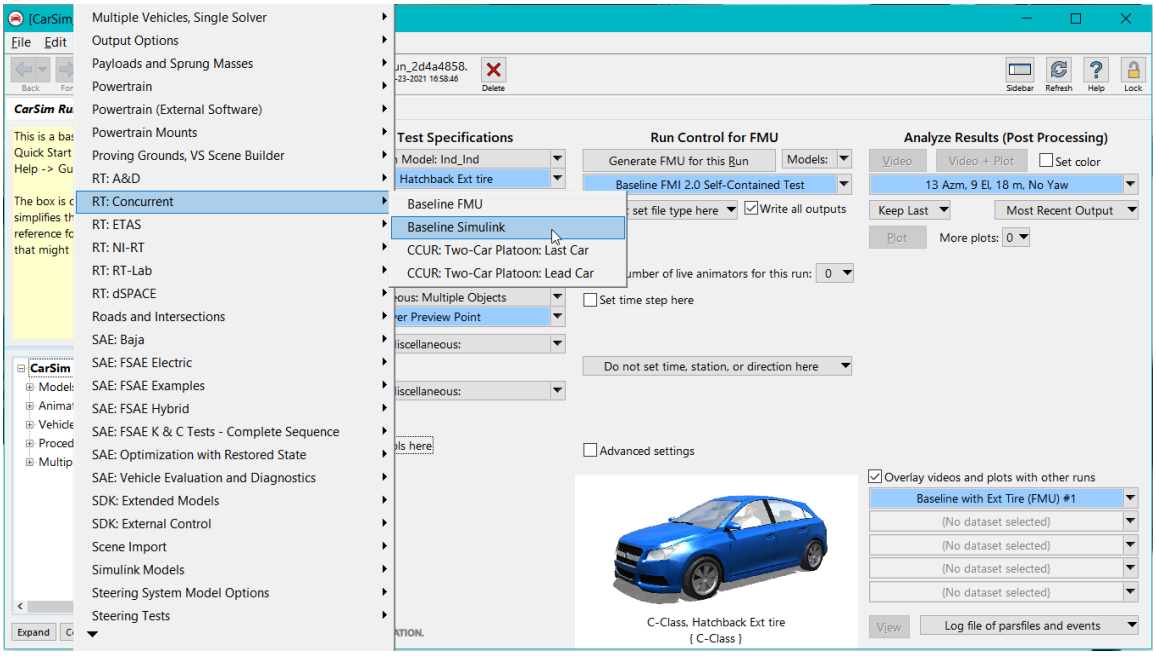

Figure 7. Concurrent Examples.

Tutorial: Run Baseline Simulink Example

The database installed has example datasets that are ready to run on a Concurrent system. You need to have and turn on the Linux-RT license to run the example.

CarSim Setup

1. Start CarSim. From the **Run Control** screen, select **Datasets > RT: Concurrent > Baseline Simulink** (Figure 8).

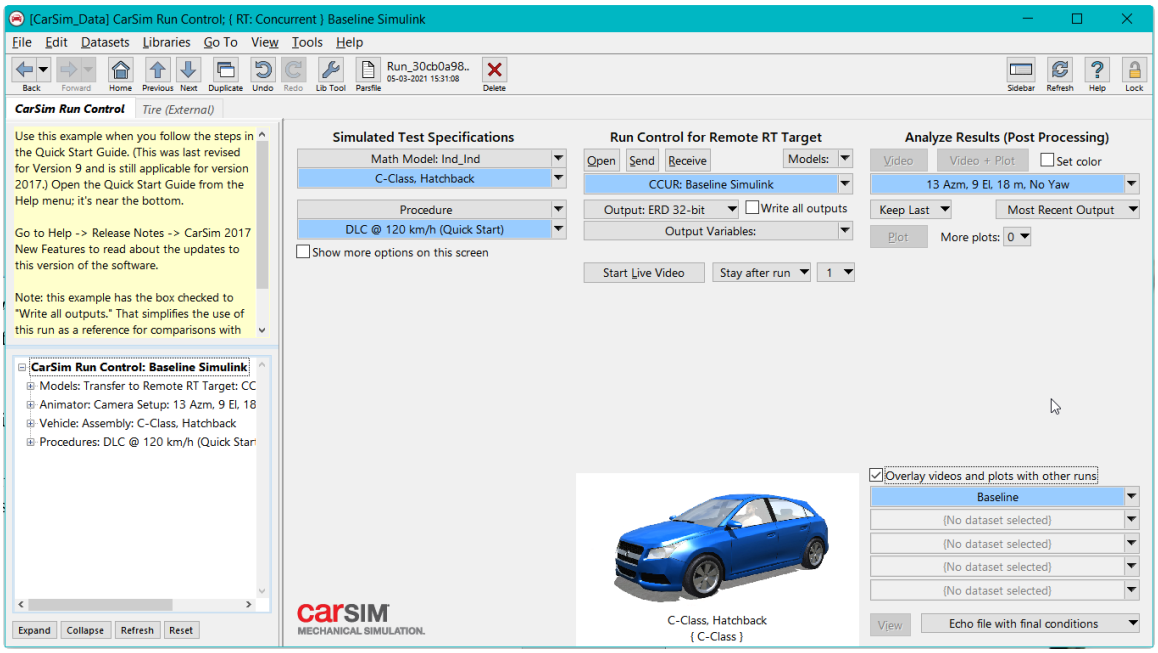

Figure 8. Run control Baseline Simulink example.

2. Use the Models drop-down list to specify the library **Models: Transfer to Remote RT Target**.

3. Select then click the **CCUR: Baseline Simulink** blue link to go to the **Models** dataset (Figure 9). Make sure `Simulink S-Function` is selected for **Use Simulink or FMU** pull-down menu.

4. Modify the target computer (RT computer) IP address and host computer IP (Windows computer) address based on your settings.

5. Go back to the **Run Control** Screen and Click the **Open** Button. This opens the Simulink model.

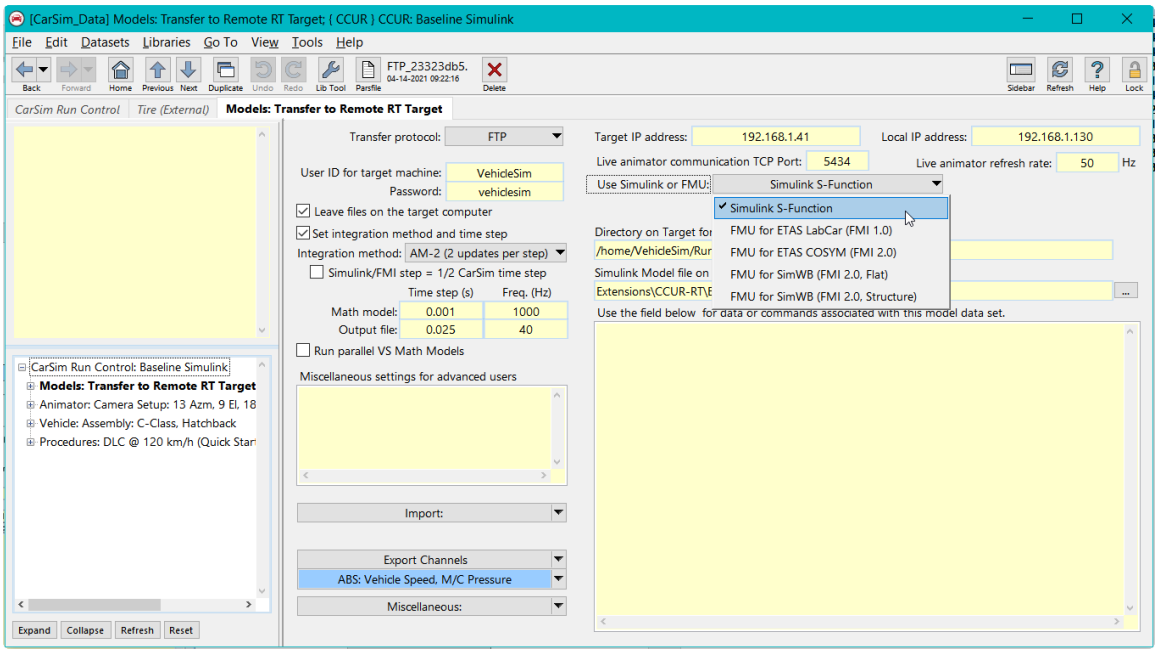

Figure 9. Models: Baseline Simulink.

SIMulation Workbench Setup

> **Note** If you are using your own Simulink model, copy `vs_sf.c` in {CarSim Prog}\Programs\HIL\`CCUR_SimWB` directory to your Simulink model location before the following steps.

1. In Simulink, click **SimWB** > **Toolkit** (Figure 10). This opens SIMulation Workbench (Figure 11). Type the real-time hostname, your SimWB username, and the password. Click **Connect** button. This will change the connection status at the button left from disconnect (the red with x) to connect (two green arrows).

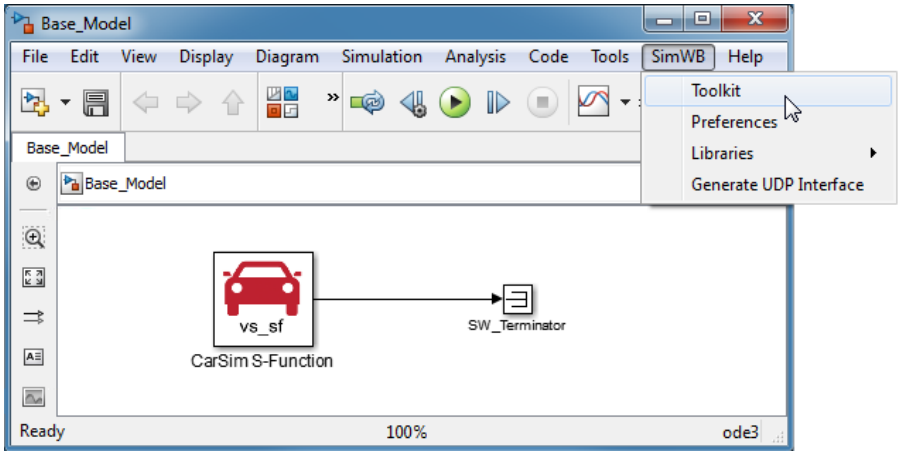

Figure 10. How to access to SimWB Toolkit from Simulink.

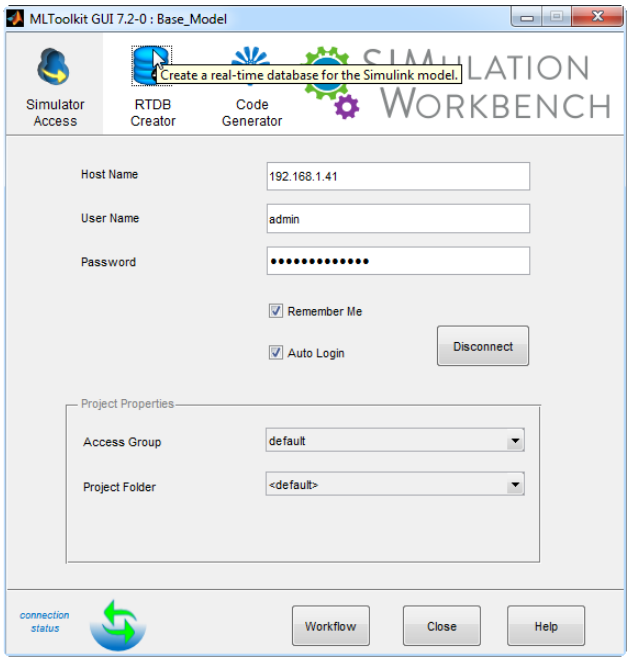

Figure 11. SIMulation Workbench pop-up window

2. Click the RTDB Creator icon. Enter the name of a real-time database. Here, we will call the database `CS_Base_SL` (Figure 12). Press the button **Create and Upload RTDB**. This will create the database from the Simulink model and upload the files to the real-time host.

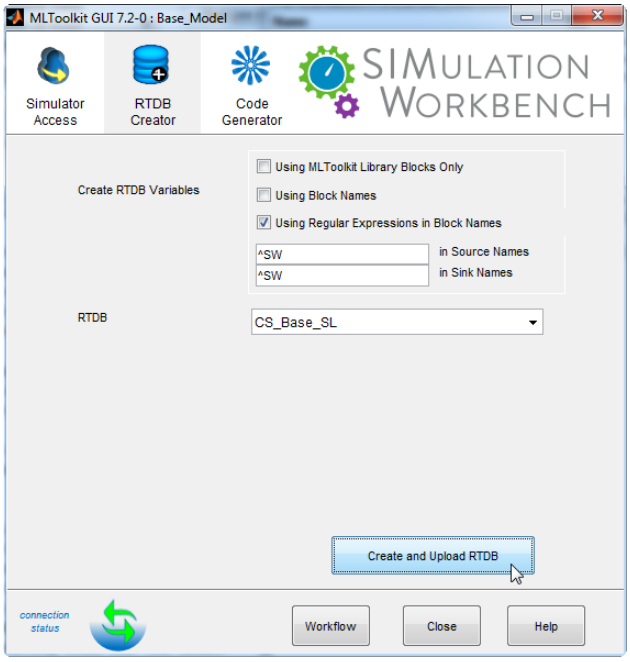

Figure 12. Simulation Workbench, RTDB Creator.

3. Click the **Code Generator** icon (Figure 13). Select the name of RTDB you have just created from the Simulink, `CS_Base_SL`. Press **Generate, Export, and Make** button. This generates code for the Simulink model, create a makefile and uses GCC compiler to

create an executable on the real-time computer. After you successfully create the executable, you can either close or minimize MATLAB.

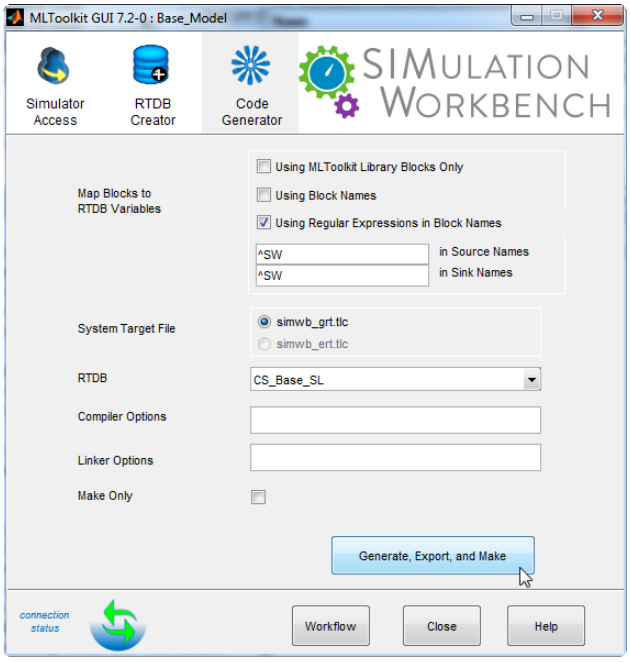

Figure 13. Simulation Workbench, Code Generator.

Note If you see the following error message when compiling the code, it indicates that you are using an old version of SIMulation Workbench.

```
*****START Create temporary SimWB Compliant Model******
*****STOP Create temporary SimWB Compliant Model*******

*****START SimWB Compliant Code Generation ************
Terminating build process.
Error using simwb.rtobj/rtwBuild (line 175)
Found CARSIM model but unable to find SIMFILE associated with it.

Error in simwbTarget>button_generatecode_Callback (line 1812)



Error while evaluating UIControl Callback

fx >>
```

To avoid this error, you should make this change:

1) Rename: `C:\Concurrent\MLToolkit\SimWB_Target\helperfiles\getcarlib.p` to `getcarlib.p.ORIG`. ("`C:\Concurrent`" is the installation folder of your SimWB.)

2) In the same directory, create `getcarlib.m` with just following two lines:

```
function libpath = getcarlib(modelname)
libpath = ''
```

SIMulation Workbench Control Center Setup

1. Open SIMulation Workbench Control Center (Figure 14). Click the **Tests…** button near the middle of the window.

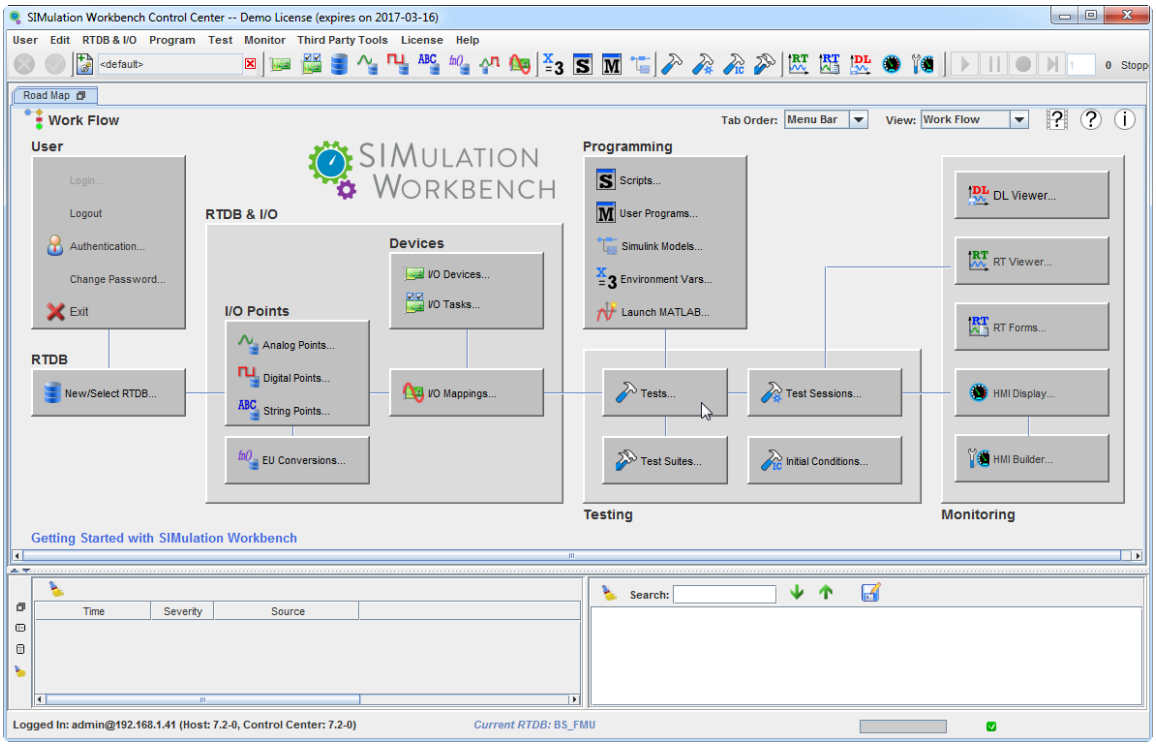

Figure 14. SIMulation Workbench Control Center.

2. After clicking the **Tests…** button, the **Tests** tab opens. From here, you can create a new test. Click the **new test** icon which looks like a paper with a green plus sign on it (Figure 15). Enter a new test name (Figure 16). Here, we will use the test name as the same name as RTDB name, `CS_Base_SL`. Select the RTDB you have created using pull-down menu. Right click in the white field and select your Simulink Model. For this example, we are using `Base_Model`. Click the green check mark icon on the top left corner to save.

3. After setting up your new test, click the Create/Run Test Sessions icon on the top (Figure 17). When the Test Session tab is open (Figure 18), enter a new test session name. We selected the name, `Test1`, for this example. Save the session by clicking the green check mark icon on the top left corner.

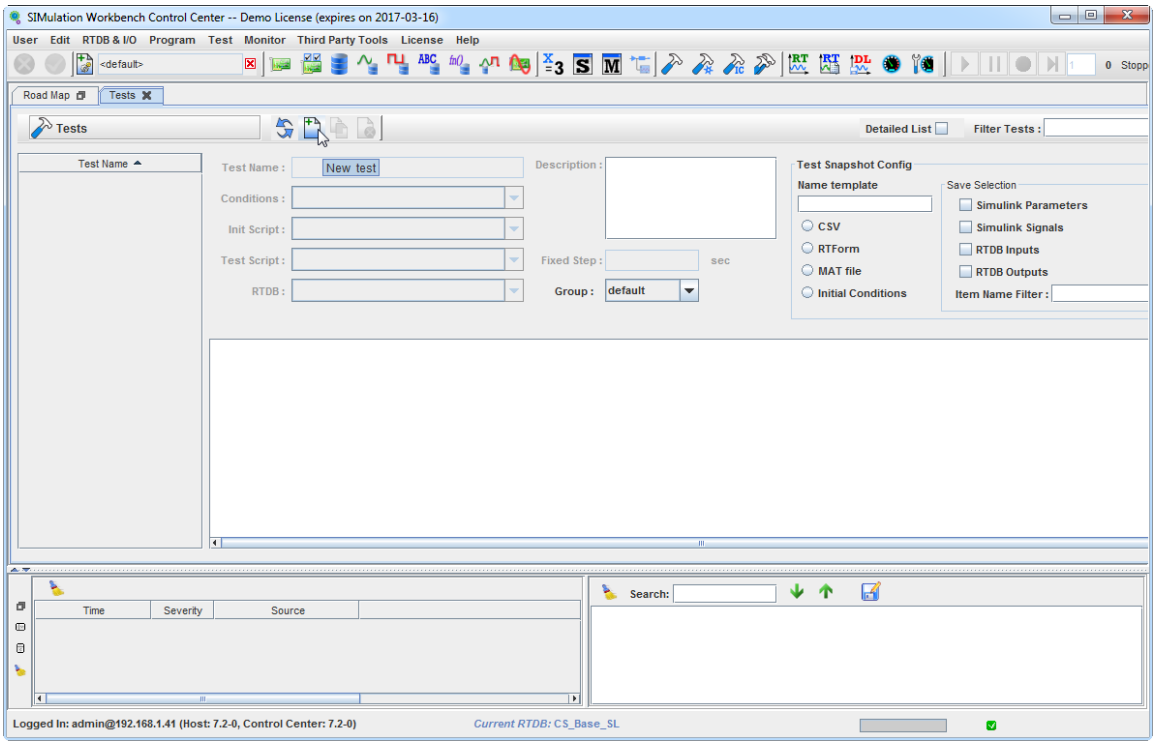

Figure 15. Simulation Workbench Control Center, creating a new test.

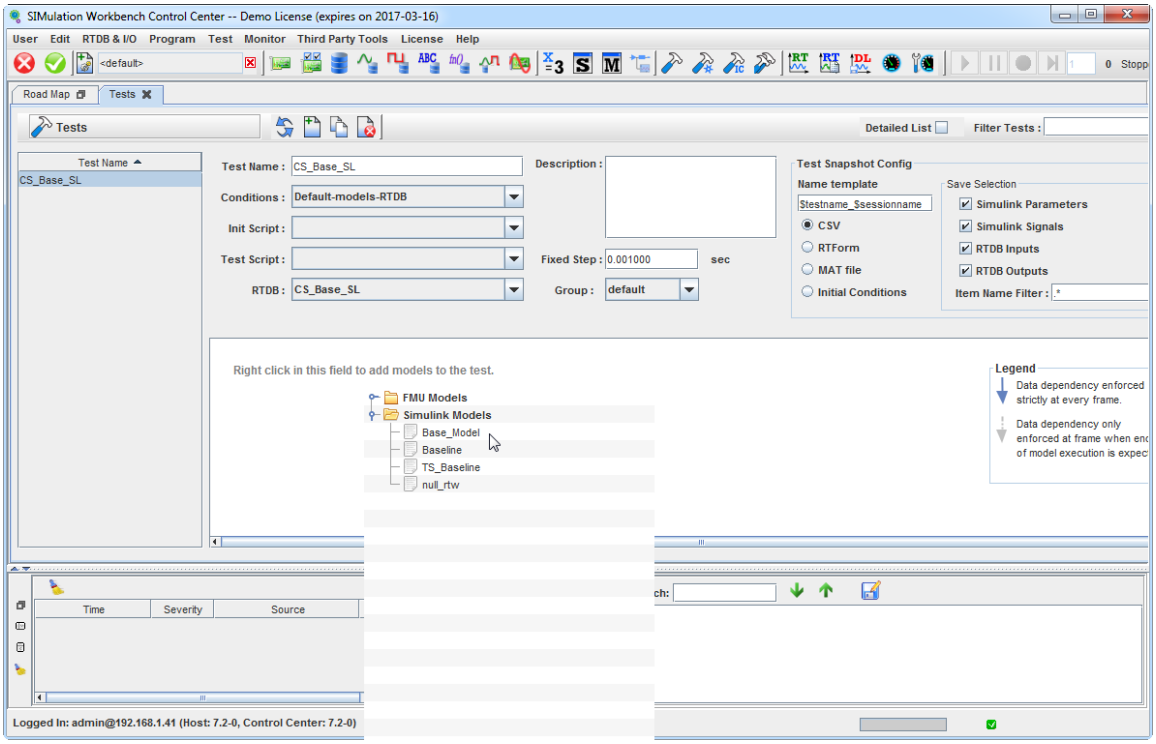

Figure 16. Simulation Workbench Control Center, setting up a new test.

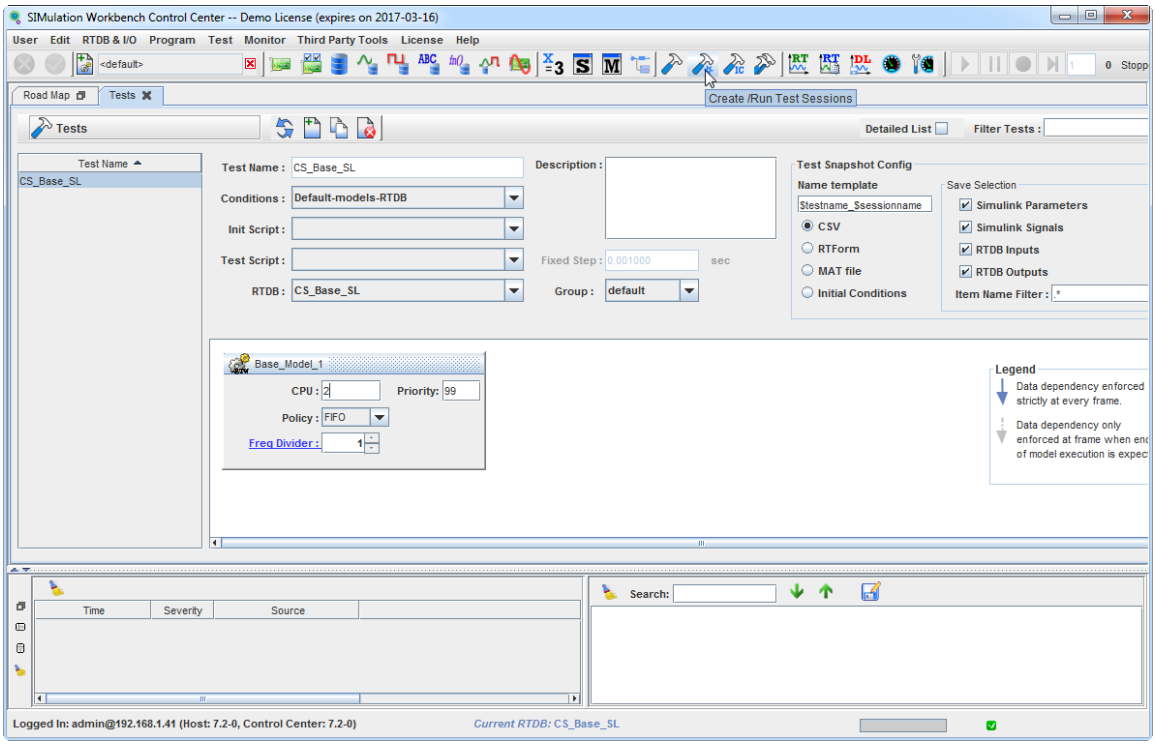

Figure 17. Simulink Workbench Control Center, click create/run test sessions icon.

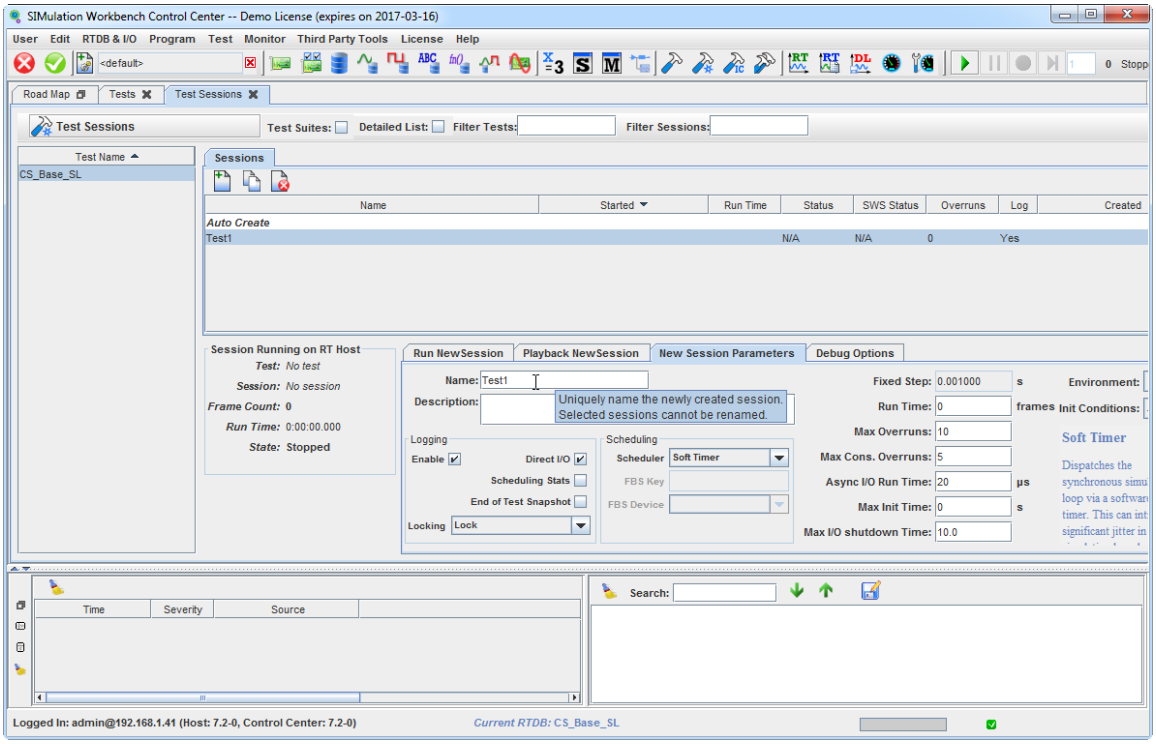

Figure 18. Simulation Workbench Control Center, Create a new session.

Run the Simulation

1. Leave the Simulation Workbench Control Center open but go back to the CarSim window. Go to the **Run Control** screen for the **Baseline Simulink** example, and click the **Send** button. This sends data for the simulation from the database to the RT target machine and launches the animator (to support live animation).

2. Go back to Simulation Workbench Control Center, still in the Test Session tab, press the play button to run the simulation (Figure 19).

3. After the run, the results are automatically transferred back to the host. If there are any problems, you can manually transfer results back to the host by clicking the **Receive** button on the **Run Control** screen. Normally, this is not necessary.

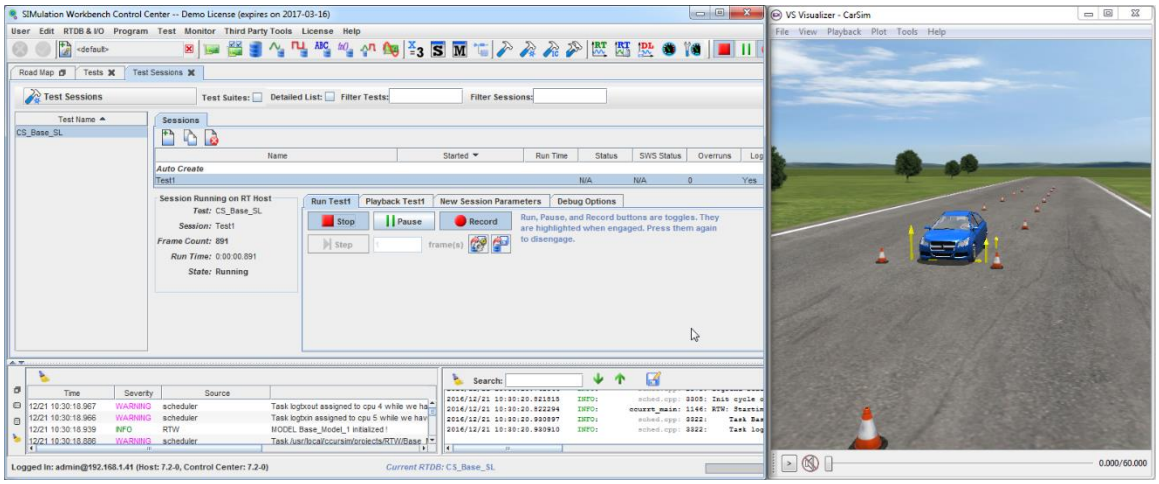

Figure 19. Simulation Running.

Tutorial: Run Baseline FMU Example

Running the baseline FMU example is similar to running the baseline Simulink example, but with a few differences.

CarSim Setup

1. Start CarSim. From the **Run Control** screen, select **Datasets > RT: Concurrent > Baseline FMU** (Figure 20).

1. Select **Models: Transfer to Remote RT Target** from the **Models** drop-down list.

2. Click the **CCUR: Baseline FMU** blue link to go to the Models dataset (Figure 21). Make sure `FMU for SimWB` is selected for the **Use Simulink or FMU** pull-down menu.

3. Modify the target computer (RT computer) IP address and host computer IP (Windows computer) address based on your settings.

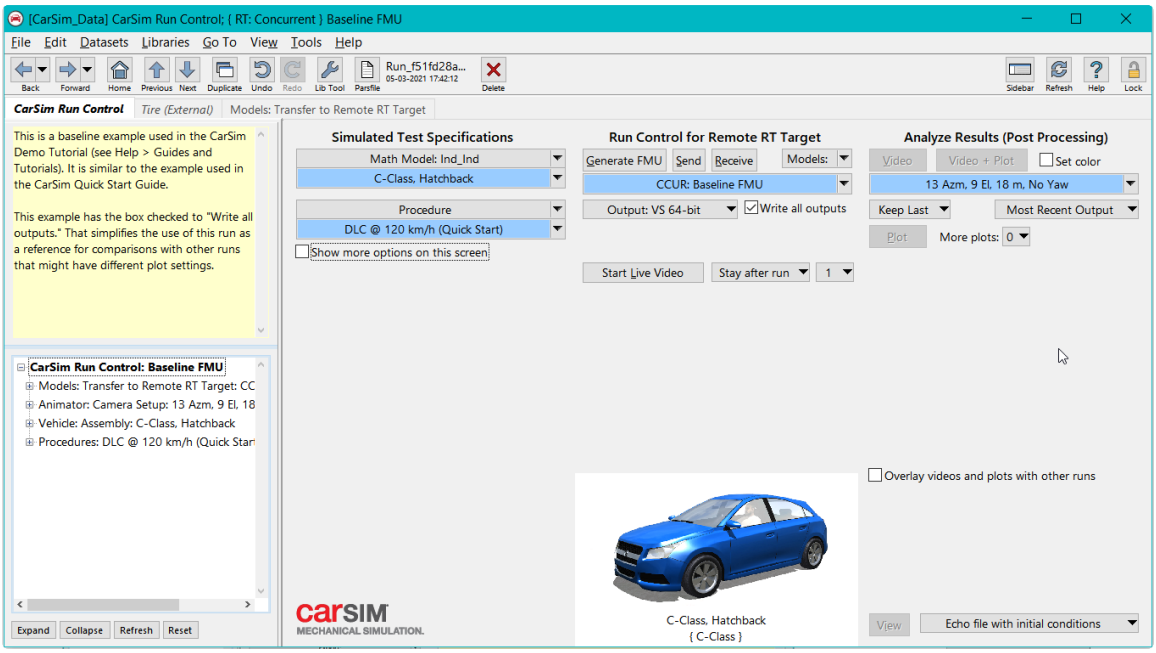

Figure 20. Run Control Baseline FMU example.

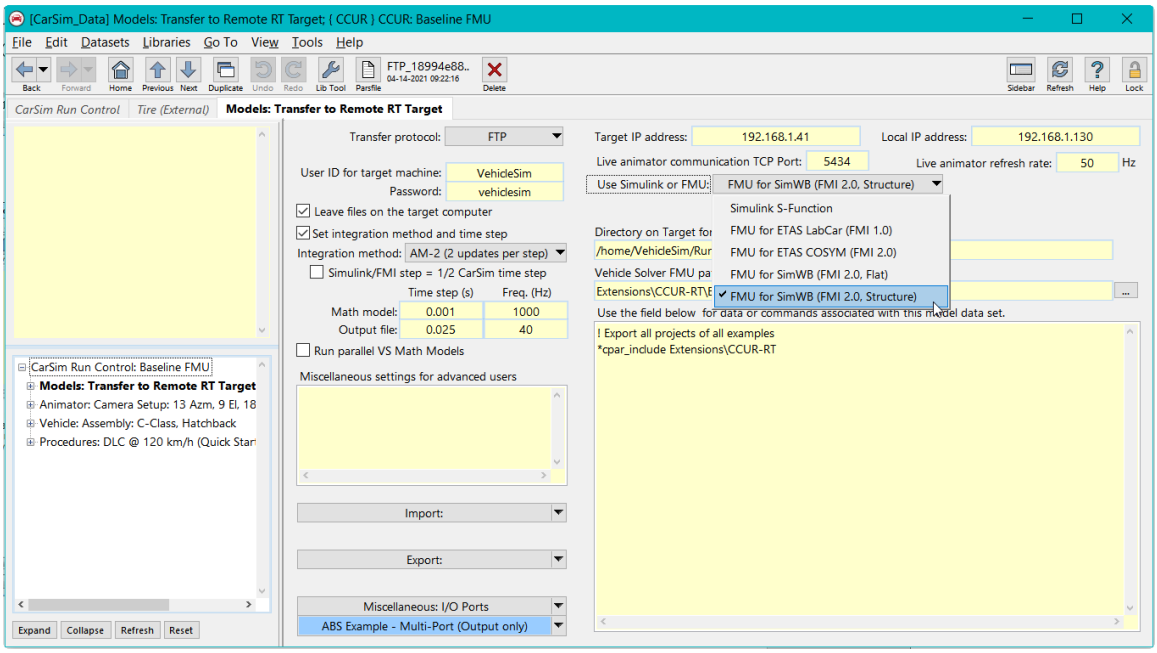

Figure 21. Models: Baseline FMU

4. Go back to the **Run Control** screen. Click the **Generate FMU** button (Figure 22).

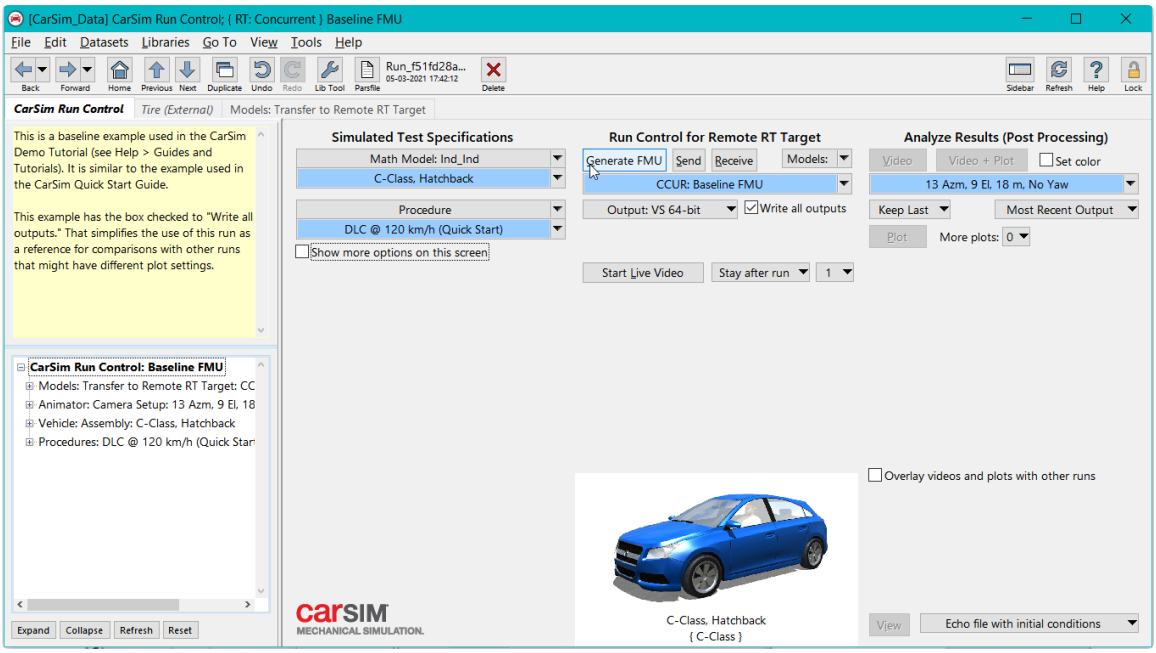

Figure 22. Run Control Screen, Generate FMU.

SIMulation Workbench Control Center Setup

1. Open SIMulation Workbench Control Center (Figure 23). Click **FMU** under the menu of **Third Party Tools**.

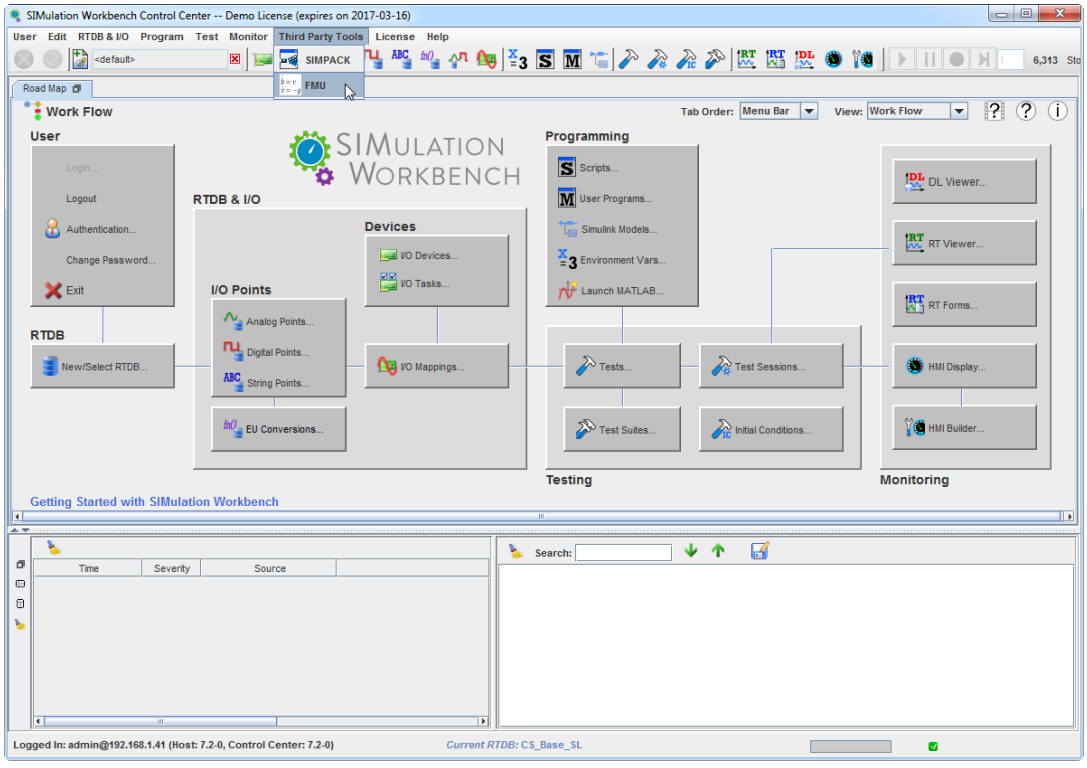

Figure 23. Simulation Workbench Control Center, selecting Third Party FMU software.

2. When the **FMU Models** tab is open (Figure 24), navigate and select your FMU file. In this example, we are using `vs_fmi2.fmu`. Click the **Parse FMU** button. This will display the inputs and outputs variables in FMU to be mapped to RTDB. For our Baseline FMU example, we only have the outputs, so no inputs are shown.

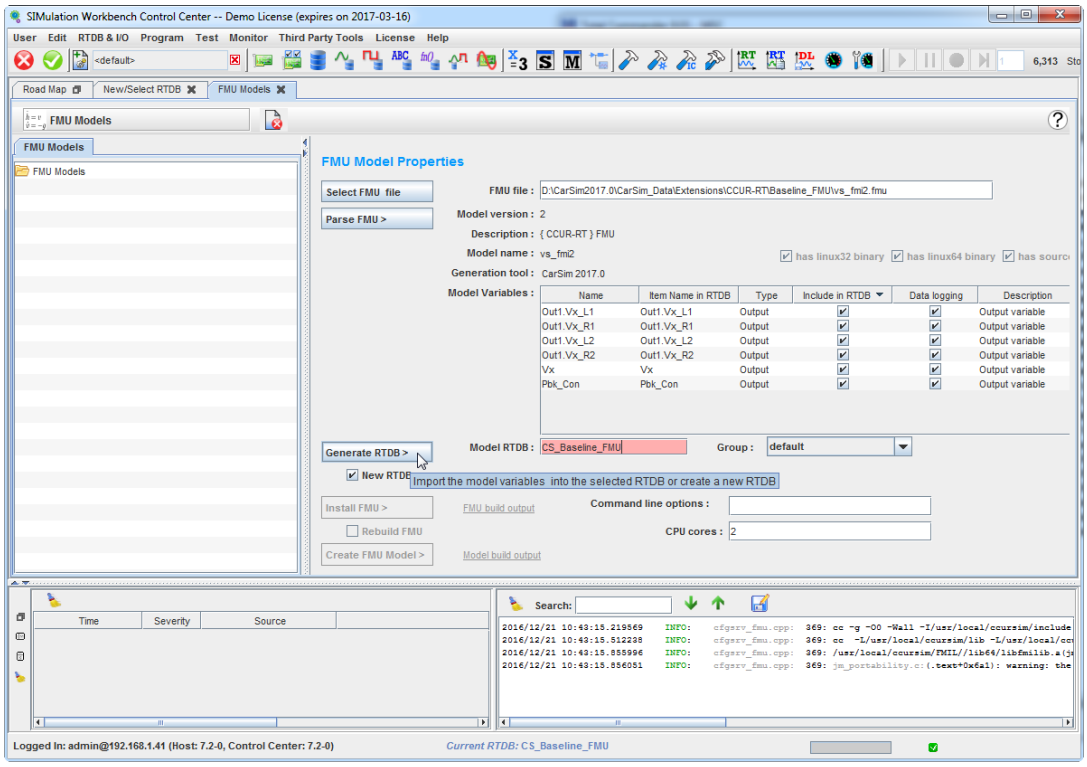

Figure 24. Simulation Workbench Control Center, Generate RTDB.

3. Click on the **New RTDB** and enter the name for RTDB. We will use the name, `CS_Baseline_FMU`. Click the **Generate RTDB** button.

4. Click the **Install FMU** button (Figure 25). Then, click **Create FMU Model** button (Figure 26). Save the model by clicking the green check icon on the top left corner (Figure 27).

5. Go back to Road Map tab in Simulation Workbench Control Center (Figure 23), click the **Test…** button near the middle of the window. In the **Tests** tab, enter a test name, `CS_Base_FMU`, and select the RTDB which is `CS_Baseline_FMU` for this example. Right click in the white field and select the FMU model (Figure 28). Save the new test by clicking the green check icon on the top left corner.

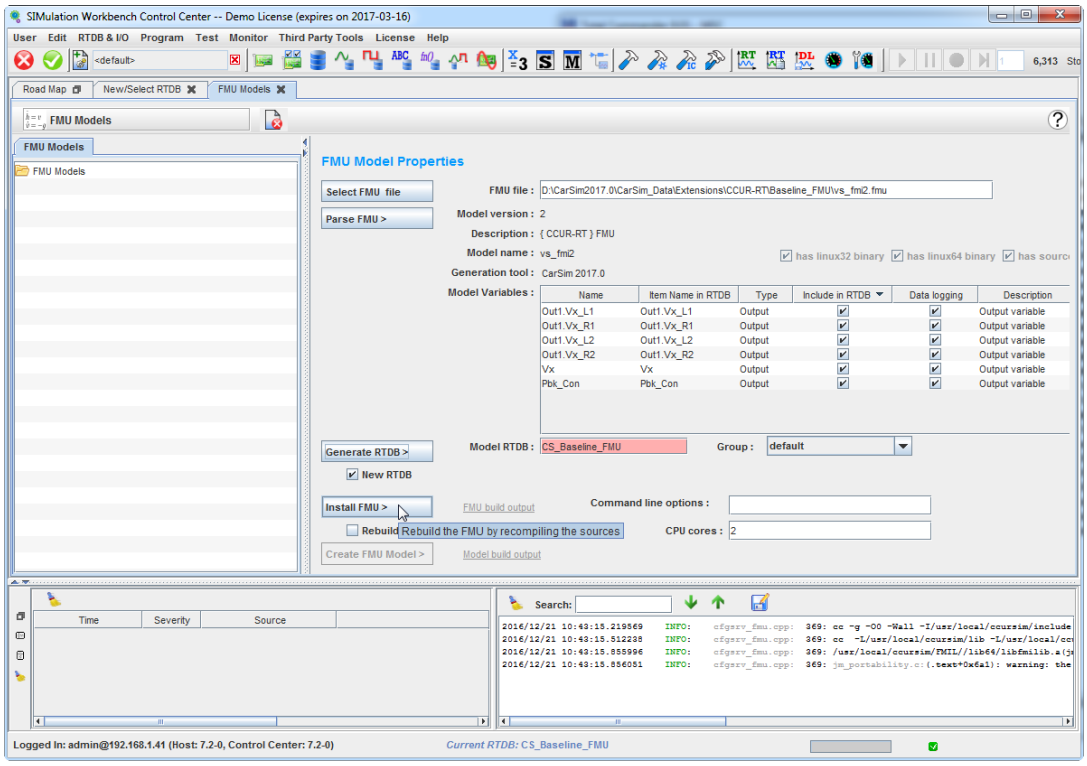

Figure 25. Simulation Workbench Control Center, Install FMU

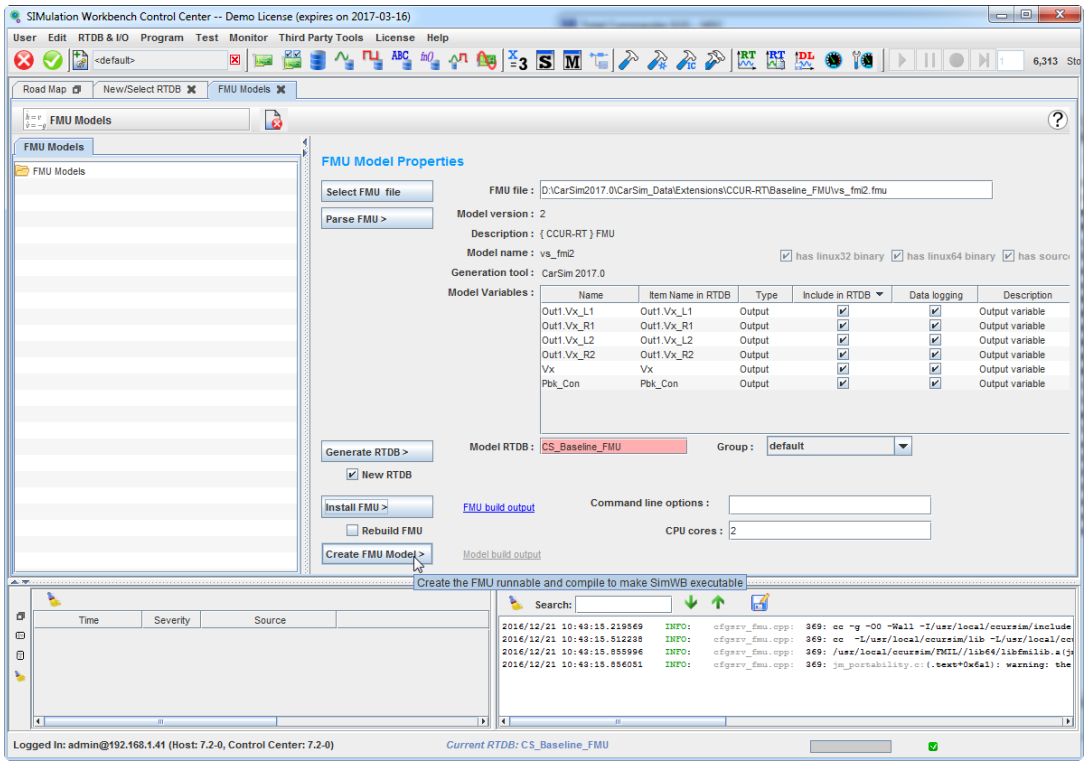

Figure 26. Simulation Workbench Control Center, create FMU Model.

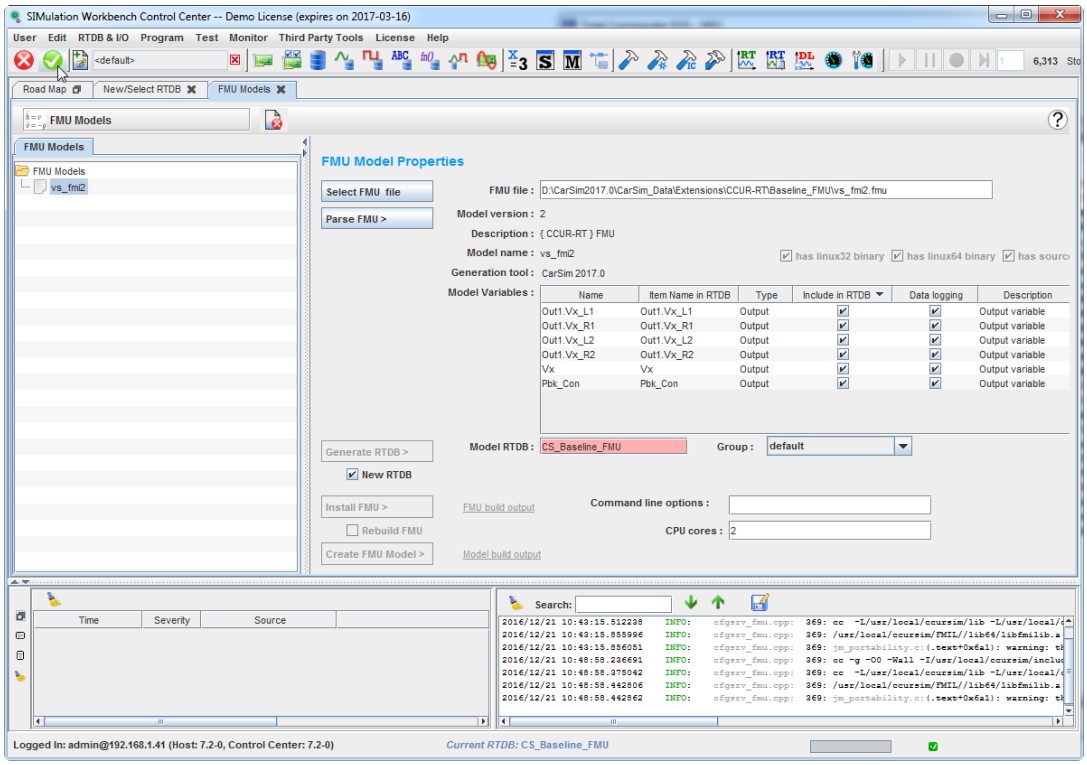

Figure 27. Simulation Workbench Control Center, after creating FMU model.

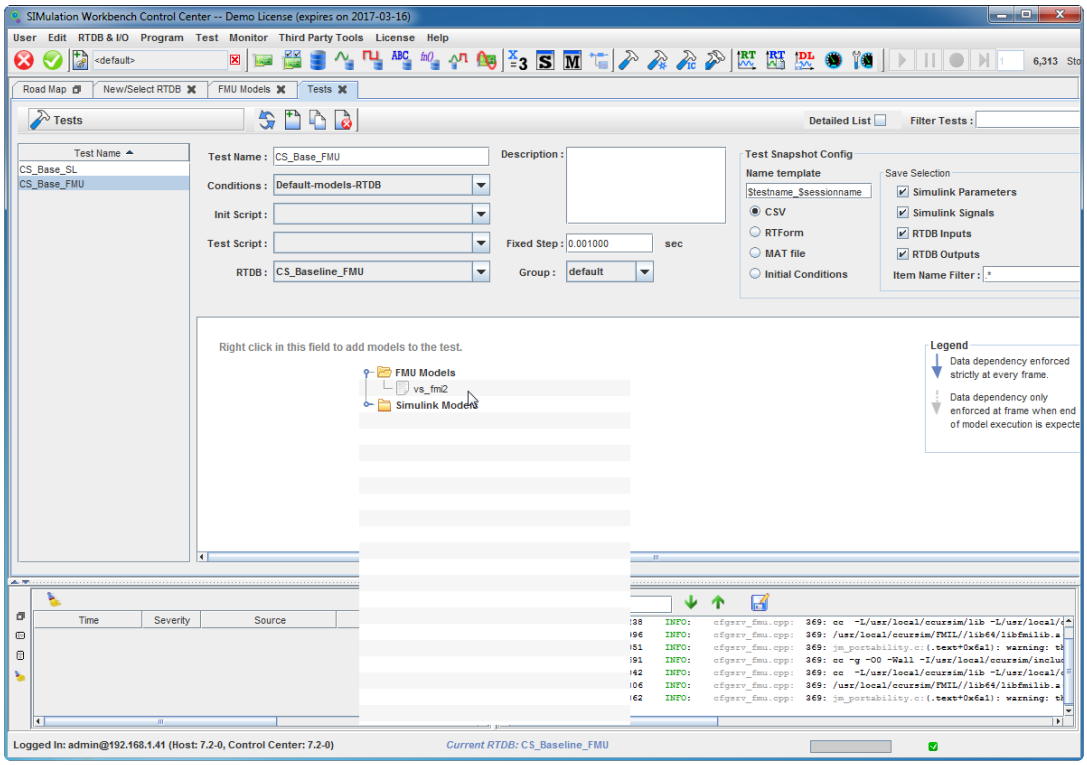

Figure 28. Simulation Workbench Control Center, create a new test.

6. Either go back to Road Map tab in Simulation Workbench Control Center (Figure 23) and click the **Test Sessions…** button or click the Test Sessions… icon on the top (see Figure 17). In the Session tab, select the test name, and enter a session name. For the session name, we use `CS_Baseline_FMU` for this example. (Figure 29). Save the session by clicking the green check on the top left corner.

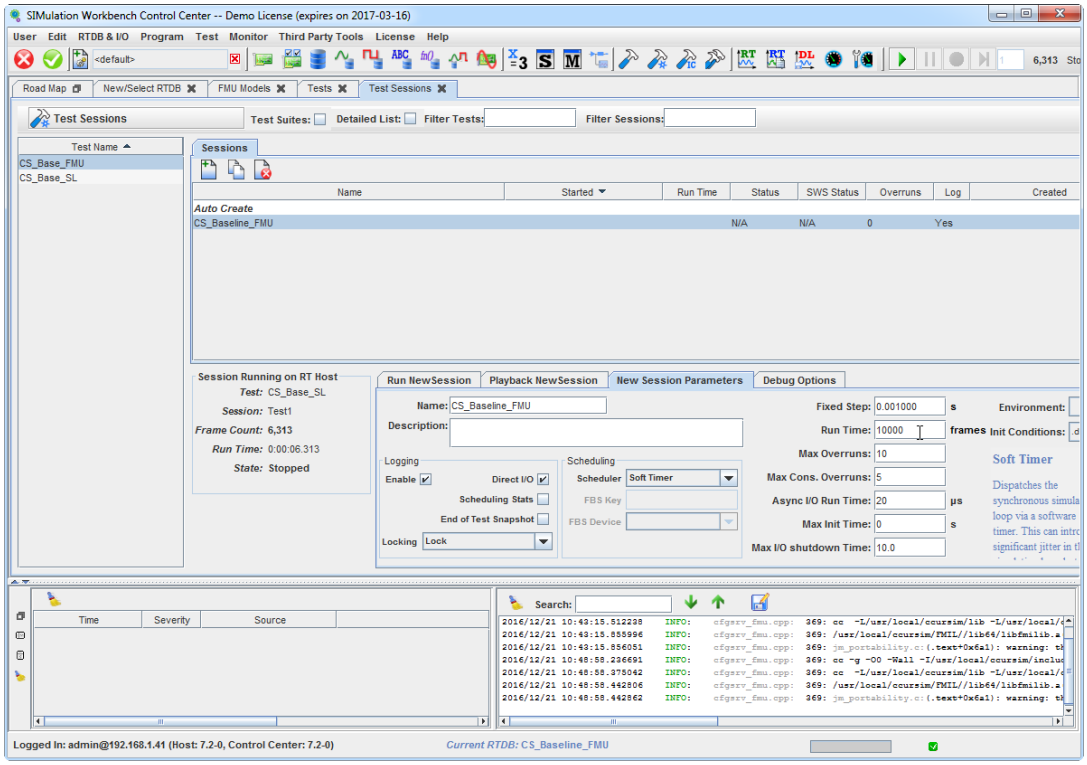

Figure 29. Simulation Workbench Control Center, create a test session.

Run the Simulation

1. Leave the Simulation Workbench Control Center window open and go back to the CarSim window. In the Run Control Screen for the Baseline FMU example, click the **Send** button. This sends data for the simulation from the database to the RT target machine and launches the animator (to support live animation).

2. Go back to the Simulation Workbench Control Center and press the **Play** button to run the simulation (Figure 30).

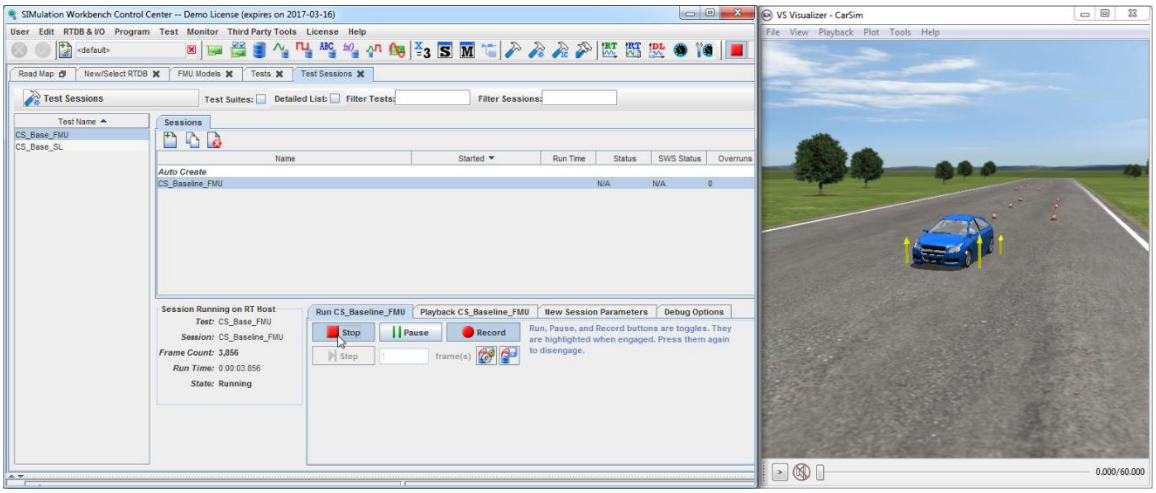

Figure 30. Run the Baseline FMU simulation.

Known Issue for FMU Simulation Model in the C file

When running an FMU simulation, you might encounter a problem which the run does not properly terminate, and the results are not saved. Work is underway to fix this problem, but in the meantime, there is a workaround:

3. Backup `fmuMain.c` in `/usr/local/ccursim/FMIL`
4. Modify the `fmuMain.c` as shown below:

In the file fmu.c:

```
                :
                :
#include <sys/types.h>
#include <sys/param.h>
#include <signal.h>   // new insert line

#include "schedutils.h"
                :
                :
```

In the function main:

```
int main(int argc,  char * const argv[])
{
    char                        *s;
    char                        cpuS[100];
    int                         i;
    int                         cpu;

                :
                :

    while((opt = getopt(argc,argv,"C:d")) != EOF) {
        switch(opt) {
                :
                :
                :
        }
    }

    signal(SIGTERM,terminate);  // Add these 3 lines
    signal(SIGKILL,terminate);
    signal(SIGINT,terminate);

    /* Use only the first CPU for now. We will use other CPUs when we add support
     * for multi-threaded FMUs
     */
    cpu_set_t cpuset;
```

Known Issue in “Electric fence” on the Concurrent RT system (FMU)

“Electric fence” will fail when running FMU model. We must disable the “Electric fence” in the makefile in the folder `/usr/local/ccursim/projects/FMU/` on the RT computer. Backup original makefile and modify it as:

Original line in the makefile:

```
$(MODELNAME): $(objects)
  $(CC) $(LDFLAGS) $(LIBDIR) -o $@ $(objects) $(LIBRARIES) -lccur_rt -ldl -lrt -lm $(FMILIB) -lefence
```

Remove **-lefence** like this:

```
$(MODELNAME): $(objects)
  $(CC) $(LDFLAGS) $(LIBDIR) -o $@ $(objects) $(LIBRARIES) -lccur_rt -ldl -lrt -lm $(FMILIB)
```

Save makefile and rebuild FMU model in SimWB.

Running Parallel Solvers Example Models

This section covers the settings that differ between running a Parallel Solvers example on Concurrent rather than on Windows. For more information regarding the standard setup for a Parallel Solvers example, refer to the 'Running with Parallel Solvers' help document. Running Parallel Solvers models on Concurrent is supported for both Simulink and FMU models.

From the Parallel Solvers screen, use the drop-down menu ① to select CCUR SimWB as the Run platform.

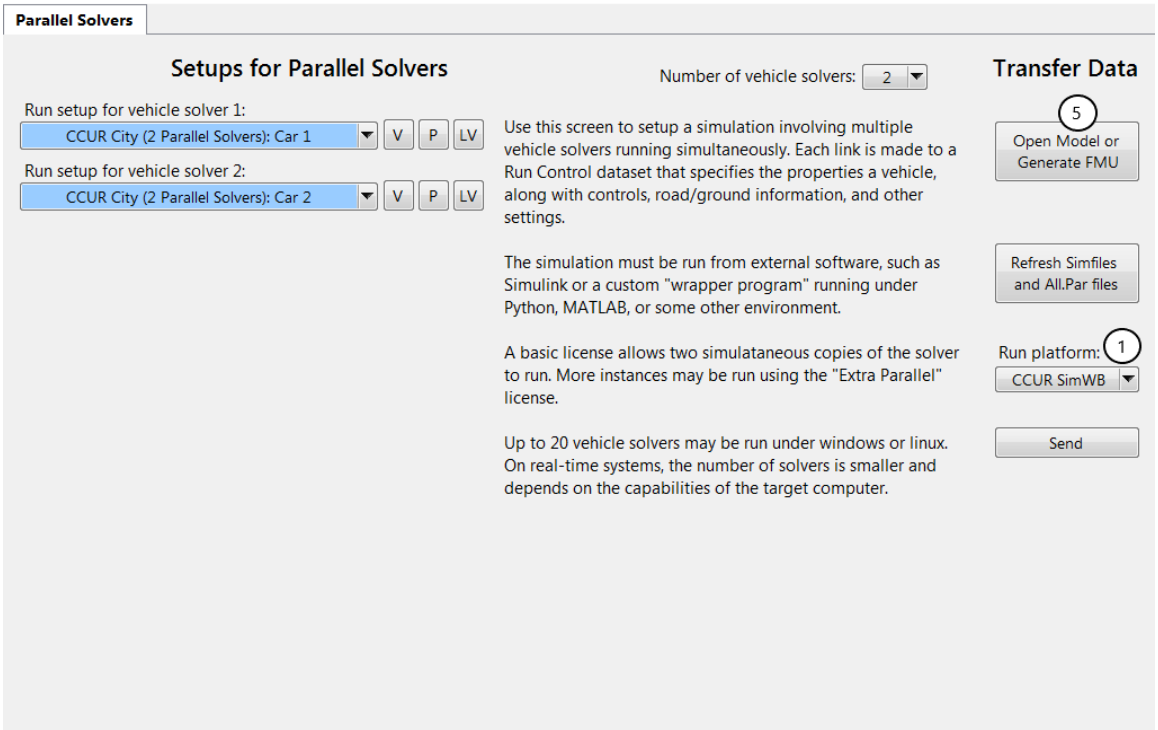

Figure 31. CCUR SimWB Parallel Solvers Screen Setup.

The 'Open Model or Generate FMU' button will generate the appropriate files depending on whether the example has been created referencing Simulink Models or FMU files.

Generate RTDB using Simulink Models

Create a project directory with a copy of the Baseline Simulink model (from the Concurrent Baseline DLC example) for each instance of parallel solvers. For ease of use, rename these copies appropriately for the vehicles being simulated (i.e. Car1.mdl, Car2.mdl).

Each Run control included in the Parallel Solvers screen will need a distinctly defined 'Models: Transfer to Remote RT Target' dataset to specify separate TCP Ports ② and Simulink models ③ from the project directory for each parallel solver.

In each of these datasets, Check the box ④ to run multiple solvers on the target machine.

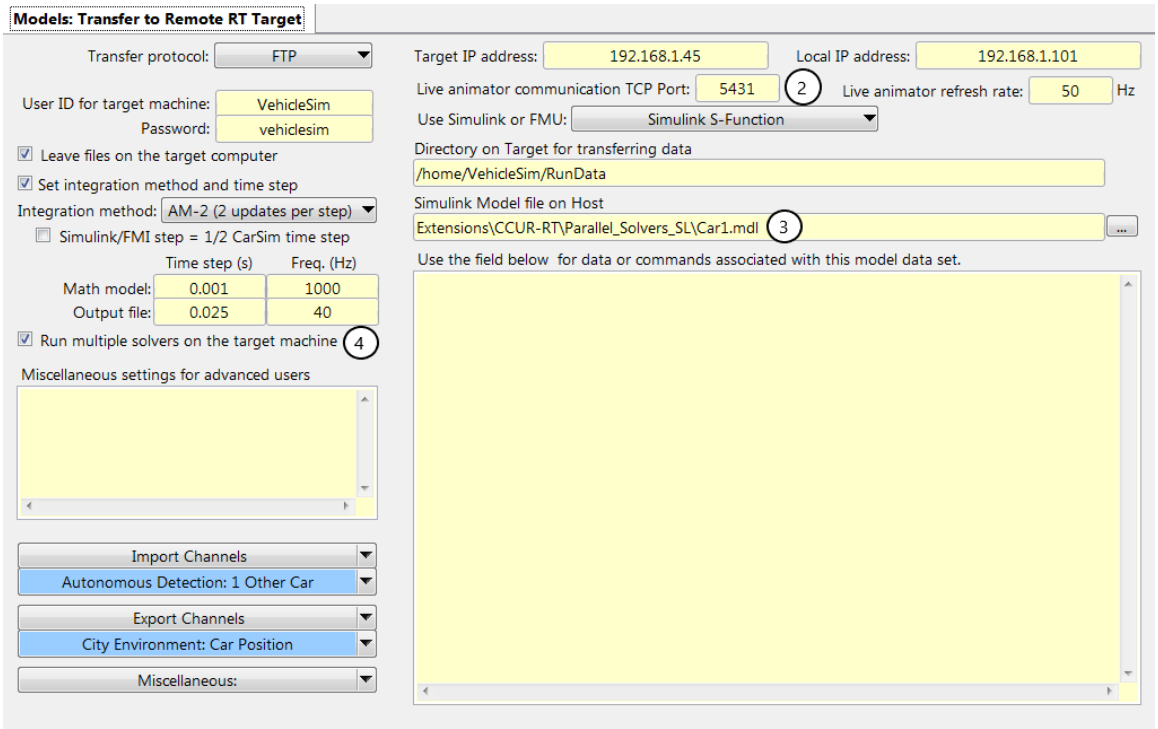

Figure 32. Model Screen for Simulink Parallel Solvers on CCUR

Navigate back to the Parallel Solvers screen. Use the Open Model button ⑤ (Figure 31) to launch Matlab and open each model.

For each Simulink Model, replace the S-Function with the Parallel Solvers S-function. Double-click this box to edit the properties:

a. Specify the appropriate parallel solver number for the Simulink Model

b. Uncheck the box specifying dSPACE SCALEXIO

c. For S-Functions that require inputs, add and connect a ground block.

   i. Rename the ground: add the prefix ‘SW_’ to follow the conventions detailed earlier in this document.

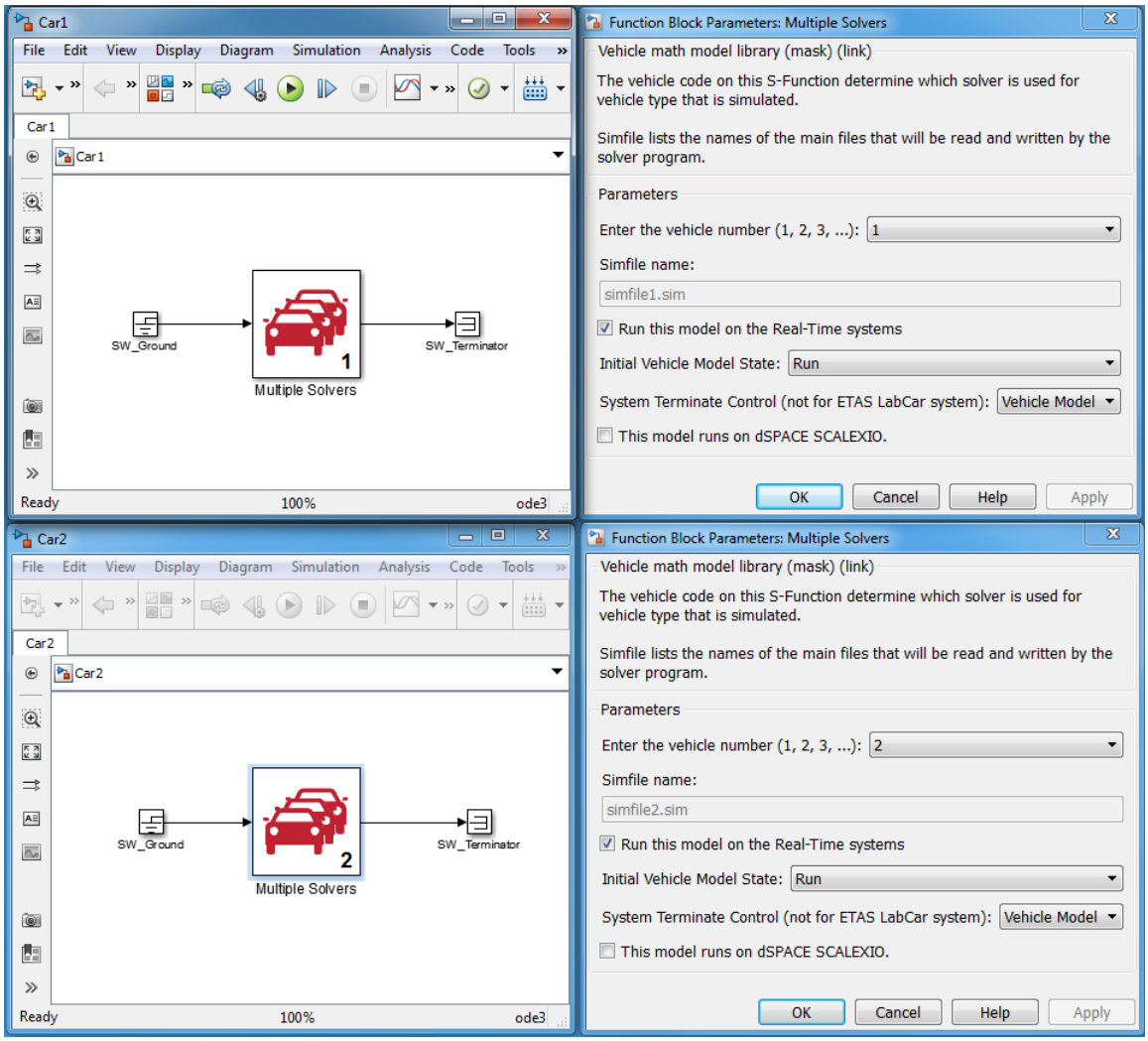

Figure 33. Simulink Setup for Two Parallel Solvers on CCUR

Starting with the first solver and proceeding sequentially:

a. Create and Upload RTDB for the first solver's Simulink model

b. Generate, Export, and Make this model

After this generates successfully, go to the next solver's Simulink model.

c. Create and Upload RTDB for this model, but do **NOT** overwrite. Select Append.

d. Generate, Export, and Make this model.

Proceed in this pattern ensuring with the Matlab command window each steps' success until all Simulink models are appended to the RTDB.

Use SimWB to connect defined IO for the Parallel Solver run.

At this point, the steps for running match that of using FMU to run Parallel Solvers on Concurrent. Proceed to subsection 'CCUR SimWB Setup.'

Generate RTDB using FMUs

Each Run control included in the Parallel Solver screen will need a distinctly defined 'Models: Transfer to Remote RT Target' dataset to specify separate TCP Ports ② and FMU files ③ for each parallel solver.

In each of these datasets, Check the box ④ to run multiple solvers on the target machine.

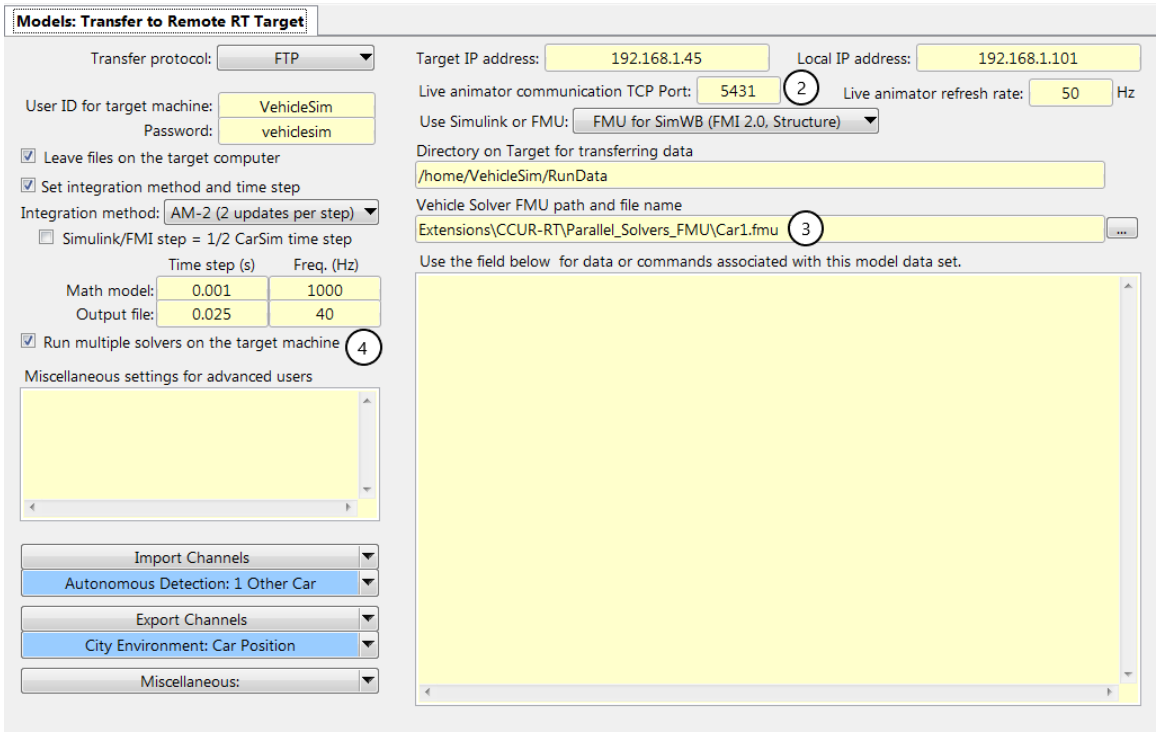

Figure 34. Model Screen for FMU Parallel Solvers on CCUR

From the Parallel Solvers screen, Generate ⑤ (Figure 31) the FMUs.

In SimWB, use the Third Party tools to select and parse the FMU of the first vehicle. Create a New RTDB and install this FMU. Ensure that this model is saved and that the IO appear for this vehicle in the RTDB. Then, select and parse the FMUs for subsequent vehicles, choosing the RTDB generated for vehicle 1 before installing. Save and ensure newly added IO after installing each FMU.

Running Parallel Solvers with CCUR SimWB

The test and session setup are similar to the standard method after generating RTDBs. Simply ensure that all the parallel solver models (FMU or Simulink) are included in the Test.

From the Parallel Solvers screen in the VehicleSim GUI, select 'Send' and open a Live Video for each run included. After the VS Visualizers load, the Run can be started from SimWB.