

Batch Matrix

The **Batch Matrix** library (Figure 1) is a tool used to create and possibly execute a set of runs with one button click. This library is accessed via the **Libraries** and **Tools** drop-down menus.

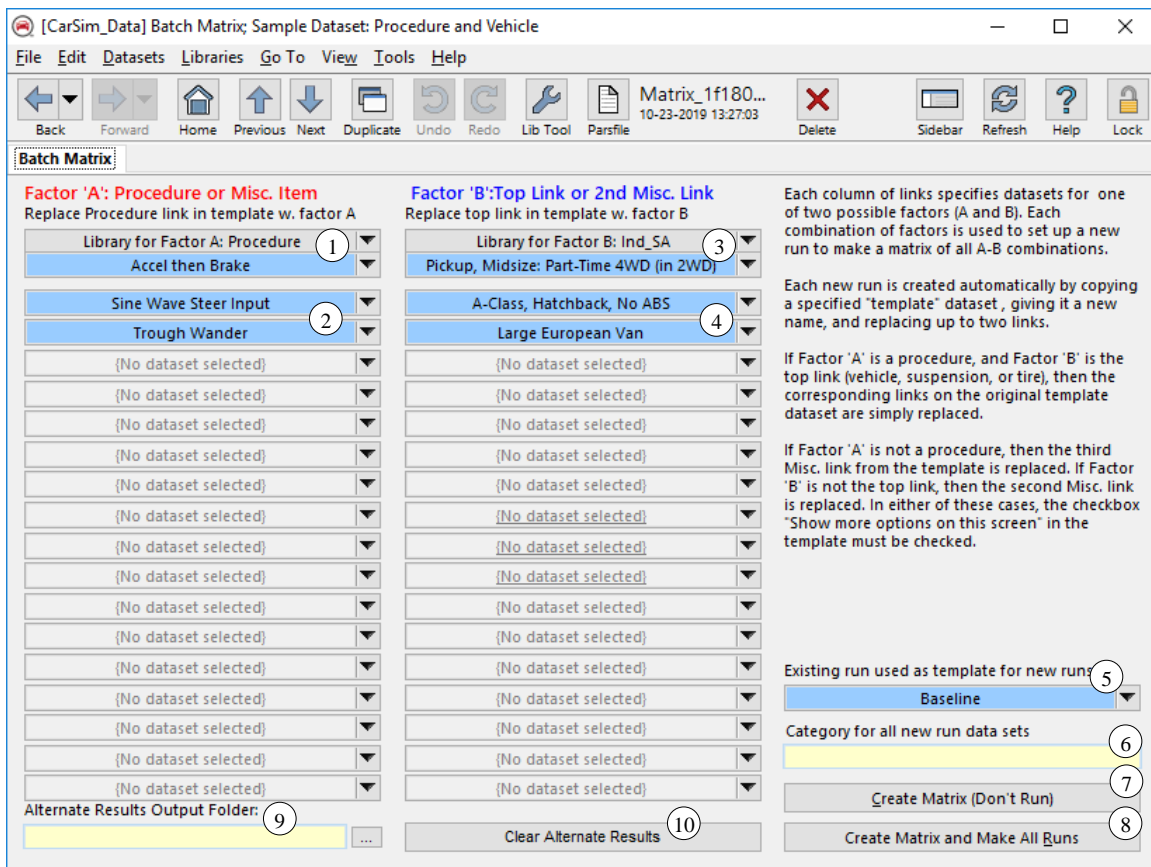


Figure 1. The Batch Matrix tool.

Overview

The Batch Matrix combines datasets from two columns of Libraries ((1) - (4)) to create a new simulation, using an existing **Run Control** dataset as a *template* ((5)). When either of two buttons are clicked ((7), (8)), each new **Run Control** dataset is created automatically by duplicating the linked *template* dataset ((5)), possibly giving it a new **Category** name ((6)), generating and applying a **Run Control** dataset title that is a combination of the names from the linked datasets for Factors 'A' and 'B' ((2), (4)), and then replacing up to two links on the new **Run Control** dataset.

The total number of new **Run Control** datasets created are based on the datasets linked on this screen. For the example shown in Figure 1, each of the three vehicles (④) will be run with each of the three **Procedure** datasets (②), resulting in $3 \times 3 = 9$ new **Run Control** datasets.

The new **Run Control** datasets reflect Factors ‘A’ and ‘B’ in the following ways:

- If Factor ‘A’ is a **Procedure** or **Tire Tester** and Factor ‘B’ corresponds to the upper left link on the **Run Control** screen — i.e., a vehicle or tire in CarSim, TruckSim, or BikeSim, or a suspension in SuspensionSim — then the corresponding datasets on the duplicated *template* are replaced with the new datasets specified on the Batch Matrix. None of the three Miscellaneous links on the new **Run Control** dataset are used.
- If Factor ‘A’ is not a **Procedure** or **Tire Tester**, the dataset(s) from Factor ‘A’ are linked on the new **Run Control** dataset(s) using the third Miscellaneous link.
- If Factor ‘B’ is not an upper left link (i.e., vehicle, tire, or suspension), then the dataset(s) from Factor ‘B’ are linked on the new **Run Control** dataset(s) using the second Miscellaneous link.

For the second and third cases, the **Run Control** checkbox “Show more options on this screen” must be checked on the linked *template* dataset to show the three Miscellaneous links, located below the **Procedure** link. If this checkbox is not checked, then when one of the buttons are clicked (⑦ or ⑧) to create the new **Run Control** dataset, an error will appear. Once the error message is cleared, the “Show more options” checkbox will have to be checked and the Batch Matrix rerun.

Note Some combinations are not supported in the Batch Matrix, e.g., runs with multiple vehicles per solver, requiring the license feature **carsimmvus**. More information is covered later in this document.

User Settings and Controls

① Factor ‘A’ library link to select almost any dataset in the database. This link serves two purposes:

- It defines the library that is used as the source of Factor ‘A’, and
- It defines the first dataset from that library to be used in the Batch Matrix. In the example shown in Figure 1, three datasets have been selected from the **Procedures** library.

If the link is made to a library that is not normally associated with the second link of the **Run Control** screen (e.g., the **Procedure** or **Tire Tester**), then the specified datasets are copied to the third Miscellaneous link on the new **Run Control** datasets.

② Datasets for Factor ‘A’ ①. Up to 18 datasets can be linked here.

③ Factor ‘B’ library link to select almost any dataset in the database. This link serves two purposes.

- It defines the library that is used as the source of Factor ‘B’, and
- It defines the first dataset from that library to be used in the Batch Matrix. In the example shown in Figure 1, three datasets have been selected from the **Vehicle: Assembly** library.

For CarSim, TruckSim, and BikeSim, Factor ‘B’ is typically a vehicle or tire. In SuspensionSim, it is typically a top-level dataset representing the suspension model. If the selection of Factor ‘B’ is made to a library that is not normally associated with the upper left link on the **Run Control** screen, then the specified datasets are copied to the second Miscellaneous link of the new **Run Control** datasets.

- ④ Datasets for Factor ‘B’ ③. Up to 18 datasets can be linked here.
- ⑤ Link to an existing dataset from the **Run Control** library to serve as a *template* for all new **Run Control** datasets created by the Batch Matrix. Neither this linked dataset nor its contents are modified by the Batch Matrix, i.e., the datasets linked for Factors ‘A’ and ‘B’ are only effected on the duplicates of the *template* created by the Batch Matrix rather than the *template* itself.
- ⑥ Field to enter a **Category** name for all new **Run Control** datasets created by the Batch Matrix. If this field is left blank, a new **Category** is not created, and all new **Run Control** datasets will be accessed directly from the **Datasets** drop-down menu on the **Run Control** screen.
- ⑦ Button to create a matrix of **Run Control** datasets, but not execute them. When this option is used, the new runs can either be executed individually from their respective **Run Control** datasets or as a group via the **Batch Runs** screen (**Tools > Batch Runs**).
- ⑧ Button to create the matrix of **Run Control** datasets and execute the runs.
- ⑨ Select an alternate directory to which the simulation results will be written for the new **Run Control** datasets created by the Batch Matrix. When this option is used, the simulation results will be written to the specified directory instead of the **Results** folder in the current product database (e.g., C:\AlternateResults vs. CarSim_Data\Results). Leave the field blank to save results to the default **Results** folder in the current product database.
- ⑩ Delete all the contents from the Alternate Results Output Folder specified in ⑨.

Note When ⑩ is used, any simulation results that have been created and saved in the Alternate Results Output Folder (⑨) are deleted, however the new **Run Control** datasets created by the Batch Matrix from the linked *template*, Factor ‘A’, and Factor ‘B’ are not affected. Therefore, the simulation results can be regenerated by rerunning the simulations either individually via the respective **Run Control** screens or as a group via the **Batch Run** library (**Tools > Batch Runs**). If the simulations are rerun from the Batch Matrix screen (⑧), the **Run Control** datasets consisting of Factors ‘A’ and ‘B’ are recreated.

Limitations and Special Cases

Although nearly any dataset from the database can be linked to Factors ‘A’ and ‘B’, not all combinations are supported. This section of the document will cover a few of these cases and discuss the reasons for the incompatibility. If certain combination of Factors ‘A’ and ‘B’ are possible but it is not clear how to create them, recommendations are made regarding their setup.

Note	This section is intended to provide some guidelines regarding commonly encountered limitations and is not intended to be an exhaustive list. If you are trying to set up a Batch Matrix and either cannot locate the appropriate library options or the Batch Matrix does not create and execute the runs correctly, please contact Mechanical Simulation or your local Agent.
-------------	--

Multiple Vehicles with a Single VS Solver

This is not supported in the Batch Matrix. The option to use Multiple Vehicles with a single VS Solver requires the vehicles to be set up via a Generic library (e.g., **Generic Data Group**), and this Generic dataset needs to be linked using the upper left link on the **Run Control** screen instead of a **Vehicle: Assembly** or **Vehicle: Loaded Combination**. When a Generic library is selected as Factor ‘B’ on the Batch Matrix, the Generic datasets will be linked to the second Misc. link on the new **Run Control** screen rather than the upper left link.

Steering

Data that is linked on the Batch Matrix screen for Factors ‘A’ and ‘B’ must be compatible with each other. One example of an incompatible data combination is linking a vehicle that uses the Virtual Steering Axis and the Steering System library that is intended for non-Virtual Steering Axis vehicles. If such a combination of libraries and datasets is linked on the Batch Matrix and the new **Run Control** datasets are created, the simulations will either terminate early due to a numerical instability, or an error message will appear indicating that certain keyword(s) are not installed. When in doubt as to which datasets can be linked to certain vehicles, locate a shipping example for the vehicle or subsystem of interest, and then see which subsystem libraries can be linked to that vehicle. It is those combinations that can be used in the Batch Matrix.

Suspension

To create a Batch Matrix consisting of vehicles and suspension data (e.g., springs and dampers), one of the Generic libraries will be selected for Factor ‘A’ (e.g., **Generic Data Group**), while Factor ‘B’ will consist of the desired vehicles. Within the **Generic Data Group**, we will specify the suspension datasets and designate their positions on the vehicle using the relevant parameters for `iunit`, `iside`, and `iaxle`.

Note	If you are unsure which vehicle index parameters to use, open the Echo file with Initial Conditions for the <i>template</i> dataset and search for these indexed parameters. Since the <code>iunit</code> parameter only applies to vehicles
-------------	--

with trailers (CarSim and TruckSim), ensure that the *template* dataset uses trailers to see this indexed data.

To see the full list of indexed parameters, use the Misc. yellow field at the bottom center of the **Run Control** screen and type: `opt_echo_all_pars 1`. Run the math model, open the Echo file with initial conditions, and look in the Model Parameters section.

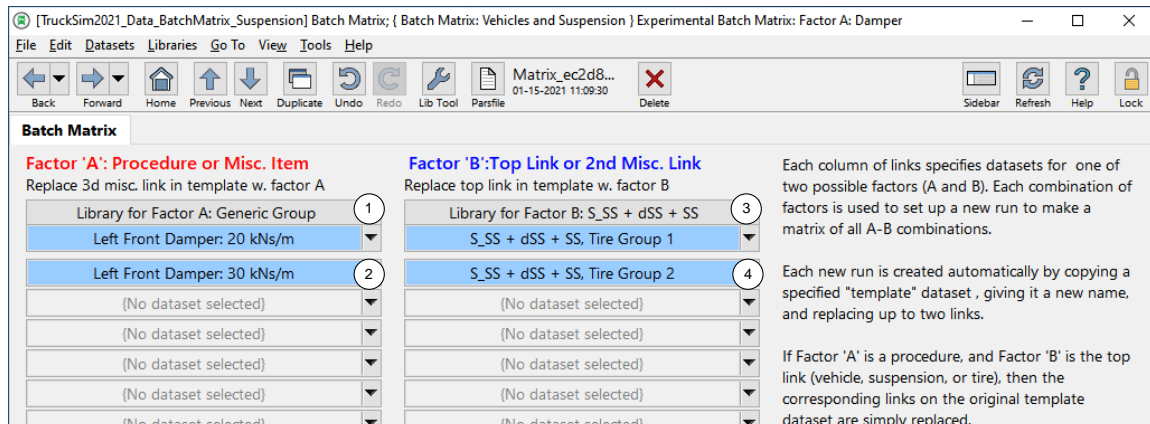


Figure 2. Batch Matrix for vehicles and dampers.

Figure 2 shows a Batch Matrix set up with two TruckSim vehicle combinations ((4)) and two dampers ((2)), yielding $2 \times 2 = 4$ new **Run Control** datasets.

Factor 'A' consists of two datasets from the **Generic Data Group** library ((1)), and within those datasets are the relevant damper datasets (Figure 3). The vehicle index parameters `iunit`, `iaxle`, and `iside` are also set. As shown in Figure 3, since only one damper dataset is linked and the values for the indexed parameters are all set to 1, this damper will be applied to the lead vehicle unit (`iunit 1`), on the first axle (`iaxle 1`), and on the left side (`iside 1`). Note that these parameters must precede the datasets to ensure the data gets applied to that position on the vehicle.

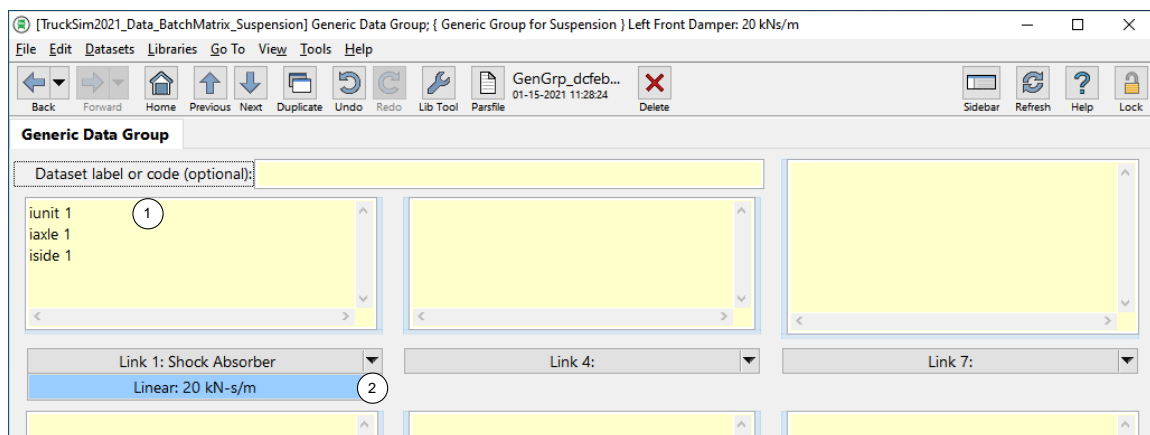


Figure 3. Generic Data Group with a damper for a TruckSim Vehicles.

Vehicles and Tires

To create a Batch Matrix consisting of vehicles and tires, one of the Generic libraries will be selected for Factor 'A' (e.g., **Generic Data Group**), while Factor 'B' will consist of the desired vehicles. Within the **Generic Data Group**, we will specify the tire datasets and designate their positions on the vehicle using the relevant parameters for *iunit*, *iside*, *iaxle*, and *itire*.

Note CarSim and TruckSim support dual tires for all axle positions (i.e., they are optional in CarSim, whereas in TruckSim dual tires are always installed). If specifying vehicles and tires via the Batch Matrix, an indexed parameter for the tires' inside / outside position will also be needed: *itire*, where 1 = inner tire and 2 = outer tire.

See the note in the previous subsection regarding the setting of *opt_echo_all_pars* to see all available indexed parameters in an Echo file.

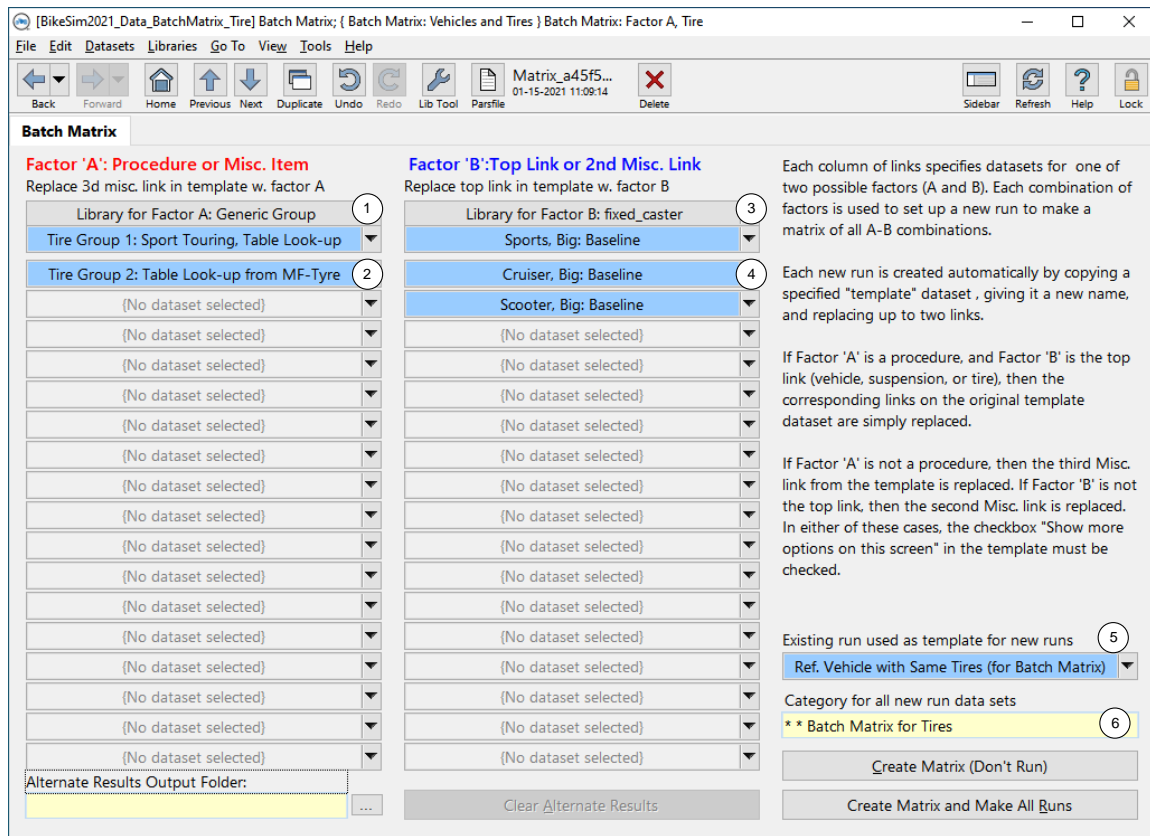


Figure 4. Batch Matrix for vehicles and sets of tires.

Figure 4 shows a Batch Matrix set up with three BikeSim vehicles ((4)) and two sets of tires ((2)), yielding $2 \times 3 = 6$ new **Run Control** datasets. The **Run Control** dataset *template* and Category used to group the new runs are specified in (5) and (6), respectively.

Factor ‘A’ consists of two datasets from the **Generic Data Group** library ((1)), and within those datasets the relevant tire datasets and `iaxle` parameters are set. As we saw with the previous example in which we created a Batch Matrix consisting of vehicles and suspension data, this example uses the `iaxle` parameter to indicate which tire will be assigned to which axle position on each of the three BikeSim vehicles. Since BikeSim only has two axle positions (i.e., front and rear), does not support trailers or dual tires, and we are not using one of the 3-wheeled vehicles, the only indexed parameter that is needed is `iaxle`. Valid values for `iaxle` will be 1 (front) and 2 (rear).

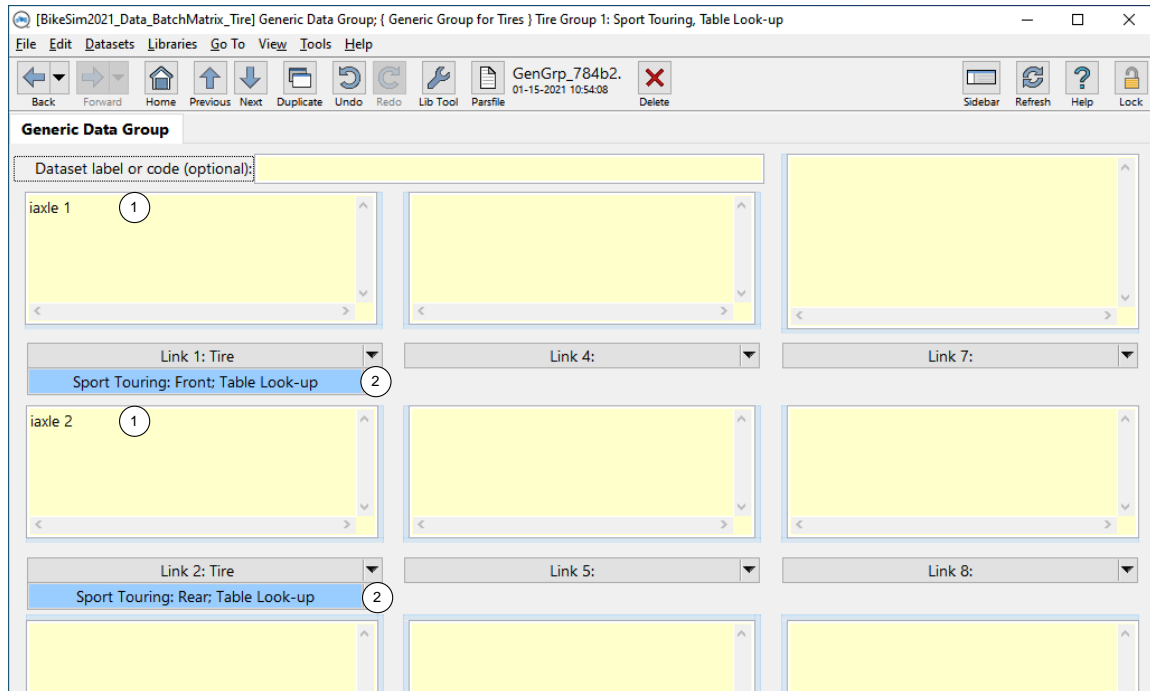


Figure 5. Generic Data Group with two tires for a BikeSim Vehicle.

Figure 5 shows the setup required to specify new tires ((2)) for the BikeSim vehicles linked to the Batch Matrix (Figure 4). The Misc. yellow fields ((1)) are used to specify the `iaxle` parameters. Note that these parameters must precede the datasets to ensure the data gets applied to that position on the vehicle.