BikeSim 2021.1 New Features

This document lists notable new features in BikeSim version 2021.1.

Database Delivery

In past versions of BikeSim, the installer application provides an example database with hundreds of examples covering basic and advanced vehicle dynamics, ADAS, VS Visualizer animation scenarios, basic and advanced uses of built-in controllers, examples using Simulink, LabVIEW, and FMI, and other examples that might be of interest to many users. For example, the 2021.0 release included a little over 200 example runs.

Examples were available separately for specialty setups, such as real-time (RT) systems involving target machines, software development kit (SDK), and others.

The delivery method has been changed for 2021.1. As with all versions in the past 15 years, the installer program installs a folder `BikeSim_Prog` with our complete program stack (`bikesim.exe` GUI Browser, VS Solver and other support libraries, VS Visualizer, VS Scene Builder, other utilities) and static resources such as animator 3D assets and Help PDF files. However, the installer no longer installs a default database. Instead, almost 300 examples are provided in over 40 Consolidated Parsfile (CPAR) archive files. Once BikeSim is installed, end users may build databases at any time, using any combination of the CPAR archives.

The new tools for installing databases are described in the next section (VS Browser). More information about the specific datasets is provided in the last section (Database, page 10).

VS Browser: Graphic User Interface (GUI)

The new options for building and managing databases are handled by the VS Browser `bikesim.exe`, using a few new features.

Recent Database Window

When opening recent versions of BikeSim, a window appears listing database folders that are known for the current version. This window (Figure 1) has been extended for 2021.1, providing access to a new tool: the Database Builder.
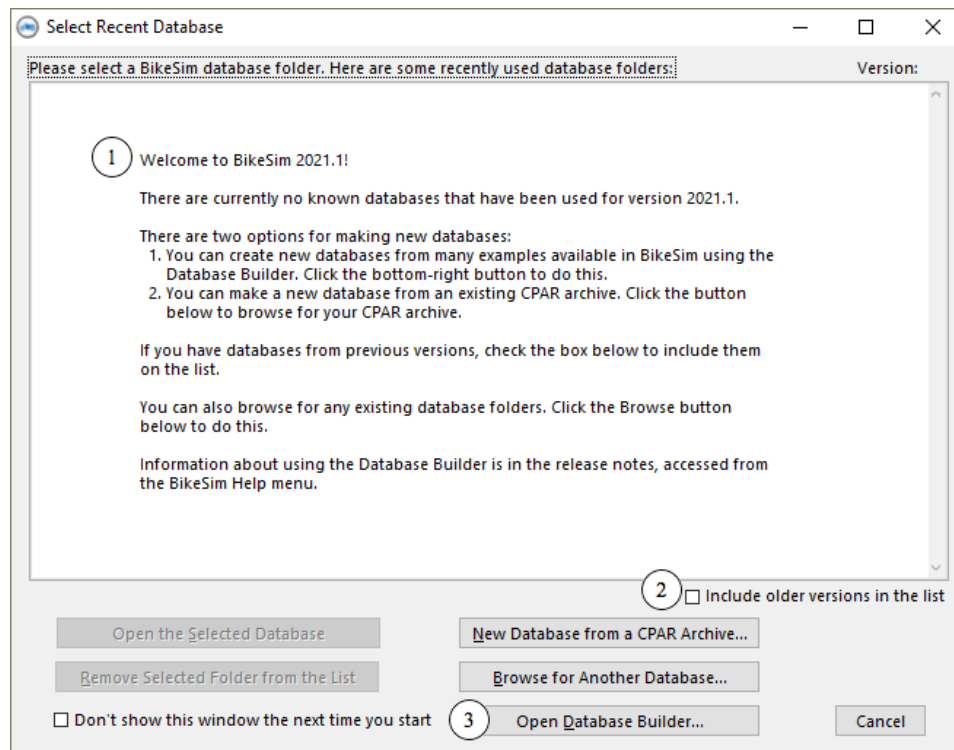

Figure 1. Recent Database window shown when opening a new version of BikeSim.

When launching BikeSim 2021.1 for the first time, the **Select Recent Database** window shows a message describing options for building a new database or accessing older databases for use in the new version ①. If you have BikeSim databases from previous versions, you can check the box **Include older versions in the list** ② and the window will show databases that were used in the previous version of BikeSim. As in past versions, any of those databases may be selected for use in the new version.

Database Builder

To access examples for 2021.1 when you start BikeSim, click the button **Open Database Builder** ③ (Figure 1) to bring up the Database Builder window (Figure 2). When a group of data is selected ①, the button **Build Database from Selected Items** ② becomes active. The **Help** button ③ opens a PDF document that describes the options.
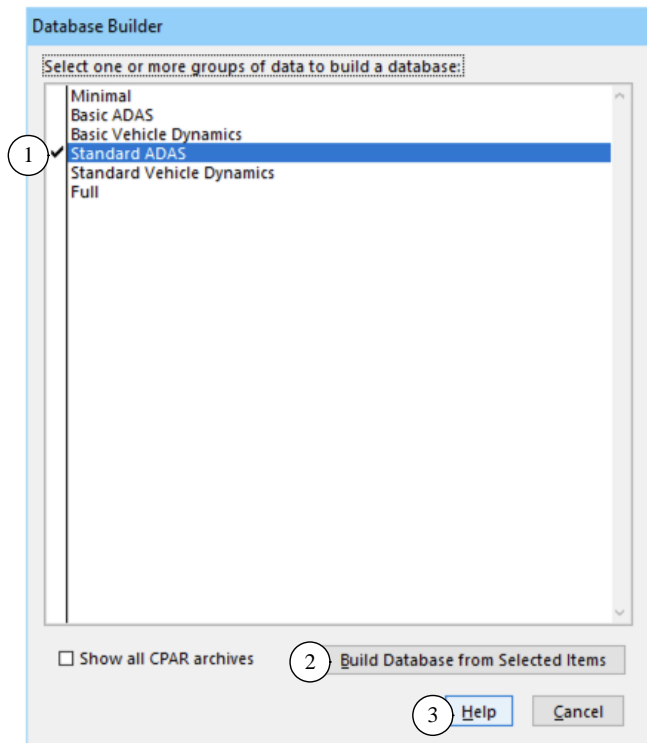
Figure 2. The database builder window showing only predefined database options.

Note	The Help document is also available from the CarSim **Help** menu in the **Release Notes** submenu.

You can access the Database Builder any time after the initial installation of BikeSim. It can be viewed whenever the **Select Recent Database** window appears. The Database Builder can also be brought up directly from within BikeSim using the **File** menu ① command **Open Database Builder** ② (Figure 3).

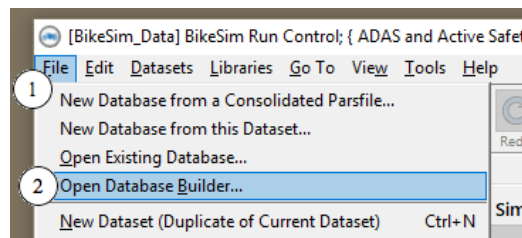*Figure 3. Use the File menu command Open Database Builder.*

Note	Three of the first four options in the **File** menu correspond directly to three of the buttons in the Enable Recent Database Window (Figure 1).

Predefined database options

Figure 2 shows that BikeSim 2021.1 provides six predefined databases. Table 1 provides a little more information about these options.

Table 1. Pre-defined database options.

Database Option	Description	Target Users
Minimal	Core examples such as the Quick Start Guide example, output options, plot setups, and preferences	Experienced users with existing databases
Basic ADAS	Minimal + most ADAS examples	New users interested in basic ADAS options
Basic Vehicle Dynamics	Minimal + examples showing vehicle properties and tests	New users interested in basic vehicle dynamics options
Standard ADAS	Basic ADAS + driver model examples, multiple vehicles, roads, proving grounds, Simulink, VS Visualizer data	Users interested in all examples relevant for ADAS applications
Standard Vehicle Dynamics	Basic Vehicle Dynamics + driver model examples, roads, proving grounds, Simulink, K&C tests, vehicle configurations, VS Visualizer data	Users interested in all examples relevant for vehicle dynamics evaluations
Full	Standard ADAS + Standard Vehicle Dynamics + Advanced VS Commands, Custom Forces and Motions, Embedded Python, Extended Models, External Control. Like example databases from older versions.	Users interested in all capabilities that do not require extra licenses or software (other than Sensors, Simulink, or LabVIEW)

Specialty Examples

None of the six basic options shown in Figure 2 include specialty examples, such as RT HIL systems, software development kit (SDK) examples, examples using external tire models, etc. To access these, and other specific categories of examples, check the box **Show all CPAR archives** ③ (Figure 4) to see the entire collection. Notice that many CPAR archives are available, shown below the six database options described in Table 1.

Table 2 lists the specialty CPAR archives that are not included in any of the six pre-defined groups. For more information, including the descriptions of all CPAR archives, please see the Help document for the Database Builder.
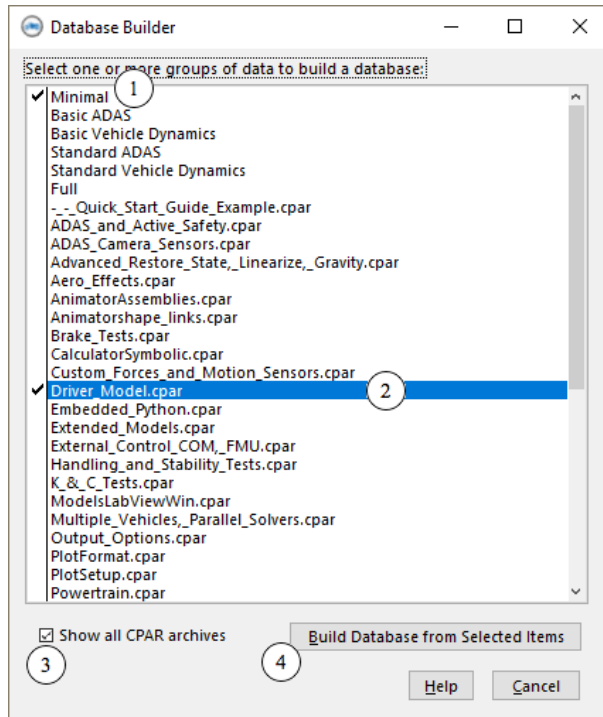
Figure 4. Display when CPARs are also shown.

Table 2. Specialty CPAR archives for standalone use or that require special licenses.

Category	Description
ADAS Camera Sensors	Use sensor memory shared by VS Visualizer
RT: A&D	Examples for A&D RT system
RT: Concurrent	Examples for Concurrent RT system
RT: dSPACE	Examples for dSPACE RT systems
RT: ETAS	Examples for ETAS COSYM and LCO systems
RT: NI	Examples for NI TS and LinuxRT systems
RT: RT-Lab	Examples for RT Lab system
SDK: Extended Models	Datasets for SDK examples that extend models or provide alternative wrappers
SDK: External Control	
Three-Wheel Motorcycles	BikeSim examples using three wheels
Tire Models (External Software)	FTire, MF-Swift

More information about specific new dataset categories is provided in the last section of these release notes (page 10).

Road: Animator Repeated Object Screen

This screen contained a feature in which objects can be made detectable by sensors. New input fields have been added which allow users to specify the `Lx_Front` and material type of the detected objects.

Tire (External) Screen

There are some enhancements and cleaning up of the Tire (External) screen to accommodate changes in the external tire models, such as discontinued TNO models that have been replaced by Siemens.

1. Removed **TNO MF-Tyre v6.2** and **TNO MF-Tyre/MF-Swift v6.2** model options as well as all corresponding model solvers, libraries and supporting files relating TNO tire model.

2. Added **Siemens MF-Tyre Only** option which distinguishes all features that can run with VehicleSim license on Windows. Any feature item appearing in **Contact**, **Dynamics**, and **Slip forces** with this MF-Tyre only model option does not require a separate license from Siemens for Windows.

3. Datasets with discontinued model options (**TNO MF-Tyre v6.2** and **TNO MF-Tyre/MF-Swift v6.2**) from prior versions can be imported and automatically assigned to either **Siemens MF-Tyre Only** or **Siemens MF-Tyre/MF-Swift** model option.

4. External road data file (RDF/CRG format) is now selectable with the Siemens model.

5. The RT Library field can specify not only the library for DS1006 (LIB) but also for Concurrent RT and dSPACE SCALEXIO (SO).

Support FMU with External Tire Model

The FMU option has been extended to include support for the external tire models MF-Tyre from Siemens and FTire from COSIN. These require that FMU 2.0 be specified.

New Time and Version Stamps Added to Parsfiles

Parsfiles written by a VS Browser are named using a 36-character unique universal ID (UUID), with a prefix associated with the library, e.g., `Run_195226c1-22a5-4584-ba9e-06e8a669c3d1` is a name that might be used for the Run Control library. This is done to support version control software that tracks creation and revisions of files based on their names.

The Parsfiles always include stamps at the end of the file beginning with '#', such as `#Library`, `#Dataset`, `#Category`, `#fileID`, and `#Product`. Three new stamps were added in 2021.1 to provide more information about time and version. Figure 5 compares the content at the end of two similar files made in versions 2021.0 (left side) and 2021.1 (right side). Both are copies of the same file: the Baseline run used for the Quick Start Guide example. The `#FileID` names differ, as expected, because they depend on UUIDs created uniquely for each file. The `#Product` stamps differ, also as expected, because two different versions were used to create and view the files.

Note that the newer version has three new stamps:

1. `#Created` identifies the date and time (obtained from the computer OS) when the file was created. This stamp will never be changed by the Browser even after updates to newer versions.

2. `#Modfied` identifies the date and time (obtained from the computer OS) when the file content was last changed. If a change is made, the time-date from the #Modified stamp is increased. However, if a change is made and then the Undo command is used (Ctrl+Z or

the Edit menu), the file contents are restored, and the time-date value is also restored to the value it had before the change was made.
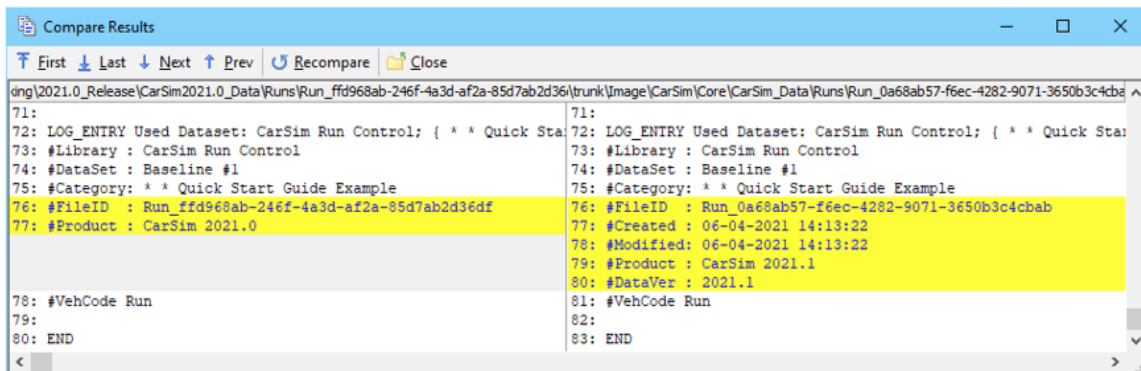

Figure 5. Timestamps compared for versions 2021.0 (left side) and 2021.1 (right side).

3. `#DataVer` identifies the version of the database. With all recent releases, the Database version (`#DataVer`) has matched the product version (`#Product`). However, this has not always been the case. (Sometime minor updates have been made for the product, but the database version was not changed.)

VS Math Models

S-L from X-Y Coordinate Transformation

Two new options are available for the potentially iterative calculation of S-L (station and lateral coordinate) from X-Y path coordinates during the simulation:

4. `OPT_SL_METHOD` (zero or one). When `OPT_SL_METHOD=1`, the estimate for station at any given iteration includes a dependence on path curvature. This can lead to more rapid convergence to a solution for station if the curvature is well-behaved. If the VS Math Model detects a failure to converge, the solution is re-attempted, this time not using path curvature. This is done automatically. When `OPT_SL_METHOD=0`, curvature is never used (the attempt using curvature is bypassed entirely), reproducing the old behavior. The default setting as of 2021.1 is `OPT_SL_METHOD=1`.

5. `TOL_SL_METHOD`, to fine-tune the accuracy of the S-L from X-Y iteration. The iteration halts when the distance is less than or equal to `TOL_SL_METHOD`, which has units of millimeters. This existed in the VS Math Model previously but was not settable by the user. The old value was 5 millimeters. The default setting as of 2021.1 is 1 millimeter.

In our testing, the new default settings (`OPT_SL_METHOD=1`, `TOL_SL_METHOD=1` mm) maintain or slightly improve performance compared to the old behavior (`OPT_SL_METHOD=0`, `TOL_SL_METHOD=5` mm), while offering improved accuracy.

Miscellaneous

1. New output variables were added to provide real time performance: `T_Real_Elapsed` (real clock time since start of run), `R_Real_Step` (real clock time used by the VS Math

Model to perform calculations for the last time step, but not counting time used by external software such as Simulink), `T_Real_Last` (total real clock time used for the last time step by the VS Math Model plus external software).

2. Users can now use VS Commands to define indexed variables and parameters with up to two dimensions. Previously, only indexed variables or parameters of one dimension could be defined by the user.

3. New output variables for position coordinates and three velocity components of the rider mass center and the total mass center of the bike (w/ and w/o rider mass) are added. The total rider C.G. combines rider's upper and lower mass and its position coordinate outputs are `Xcg_RM`, `Ycg_RM` and `Zcg_RM`, while the outputs of velocity components are `Vx_RM`, `Vy_RM` and `Vz_RM`. The total bike C.G. w/o rider combines all masses of the main frame, steer head, lower fork, swing arm and front/rear wheels and its position coordinate outputs are `Xcg_TM`, `Ycg_TM` and `Zcg_TM`, while the outputs of velocity components are `Vx_TM`, `Vy_TM` and `Vz_TM`. The position coordinate outputs for the total C.G. of bike w/ rider are `Xcg_TMR`, `Ycg_TMR` and `Zcg_TMR`, while the outputs of velocity components are `Vx_TMR`, `Vy_TMR` and `Vz_TMR`.

4. Nomenclatures for sprung mass (main frame) and intermediate coordinate system in `Run_out.txt` are changed from [`mainx`, `mainy`, `mainz`] and [`ex`, `ey`, `ez`] to [`sx`, `sy`, `sz`] and [`x`, `y`, `z`], respectively, in order to be consistent with BikeSim/TruckSim.

RT Platforms

Starting with version 2021.1, BikeSim supports COSYM v2.3 from ETAS with Simulink or FMI 2.0 for COSYM and parallel solvers application.

VS Visualizer

VS Visualizer can now compute first and second derivative estimates of data channels and use these values for both animation and plotting. See the VS Visualizer reference manual for more information.

Documentation

The following documents have been added to the **Help** menu:

1. BikeSim Database Options (Release Notes)

2. VS Table Tool Introduction (Reference Manuals)

3. VS Table Tool Usage (Technical Memos)

The following Guides and Tutorials have been updated:

4. * Introduction to CarSim (and BikeSim)

5. BikeSim Quick Start Guide

6. Borrowing a License from a Network Server

The following Deprecated Items have been updated:

7. Clutch Control (Closed Loop) Screen

8. Control: Upper Rider Position Screens

9. Path X-Y Coordinate Screens

10. Powertrain Transmission Screens

11. Speed (Closed Loop) vs Station Screen

The following Reference Manuals have been updated:

12. MF-Tyre/MF-Swift User Manual

13. VS Browser (GUI and Database)

14. VS Math Models

15. VS COM Interface

16. VS Commands

17. VS Commands Summary

18. VS Visualizer

The following Screen documents have been updated:

19. ADAS Sensors and Moving Objects

20. Animator Reference Frames

21. Animator Shapes and Groups

22. Animator Wheel Arrows and Other Indicators

23. Batch Matrix

24. External Models and RT Systems

25. Paths and Road Surfaces

26. Road Surface Visualization

27. Run Control Screen (Home)

28. Tire Models

The following Technical Memos have been updated:

29. Automating Runs with the VS API

30. Convert Simulink Model into Executable

31. Example: Extending a Model with VS Commands and the API

32. Extending a Model with Embedded Python

33. HPC Licensing

34. Parking Lot Layout and Dimensions

35. VS Solver Wrapper

The following Real-Time and DS System documents have been updated:

36. Concurrent RT Guide

37. NI RT Guide

The following Software Development Kit (SDK) documents have been updated:

38. The VehicleSim API — Running and Extending VS Solvers

39. The VS Connect API — Inter-process Data Replication via UDP

Database

Database Builder

With the new Database Builder, each category in the Run Control library is provided with a separate CPAR archive file. Many of the categories and titles of existing database were modified. If the content is otherwise the same, then the original UUID file name was kept.

In many of the categories, more runs were included to ensure that older datasets of interest would still be included. The following subsections describe significant new examples or updates of existing examples.

New and Updated Examples

- An example using the new output variables for position coordinates and three velocity components of the rider mass center and the total mass center of the bike (w/ and w/o rider mass) is added. In the animation, the C.G. positions of the total rider mass, total mass of the bike w/ and w/o rider are displaced using the new output variables. Also, the C.G. position of the main frame, steer head, lower fork, swing arm and front/rear wheels are displayed using existing output variables. The plots display the instant velocities of the rider C.G. and total C.G. of the bike, w/ and w/o rider.

- STI wrapper for FTire model is changed to be more robust and showing a proper warning message especially when an unsupported version of FTire (older than v2020-4) is used.