CSV export

On this page

• Support of Part properties

As of Version 18.4, Magic Model Analyst can export simulation results to a CSV file.

A new configuration, called CSV Export, has been added to export simulation results to a CSV file. CSV files take the following properties:

• File Name

The name of the file that the exported results are written into. The path is the same as the project directory (by default). Alternatively, a full path may be specified as D:\ExportedCSVResults.csv

- Record Time
- If true (by default), time will be recorded in the first column of the exported CSV file.
- Represents
- Specifies the Classifier represented by the Configuration.
- Value Specifies the properties of the Classifier (including nested properties) whose values will be recorded and written to the CSV file.
- Write At The End
- Records values only once at the end of execution, right before termination, rather than listening and recording value changes during the execution. When **Number of runs** is more than 1 in the main configuration, the values are recorded once in each iteration.

To export simulation results to a CSV file

- 1. Open a sample simulation project from the samples folder of Magic Model Analyst. Here, the HingeMonteCarloAnalysis sample is used.
- 2. Drag the CSV Export Configuration from the toolbox to the Simulation Configuration diagram.
- 3. Double-click the CSV Export Configuration and fill in values for File Name, Represents, and Value from its Specification window. The File Name property is the name of the file in the project's working directory. The Record Time property is for recording time in the first column of the exported CSV file. The Represents property specifies the Classifier represented by the Configuration, and the Values property refers to the values selected which will be written to the exported CSV file. Optionally, you can also set the Write At The End property to record values only once at the end of execution before termination. The following figure shows the parameters of CSV Export configuration.

Specification of CSVExport clearance									
Specification of CSVExport properties Specify properties of the selected CSVExport in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.									
🖬 🗟 🔁 💈 clearance									
elearance	🛃 🖳 🚓 ab: Properties: Stand	lard 🗸							
	File Name dearance.csv								
	Name dearance	dearance							
	Record Time 🗹 true	🗸 true							
	Represents 📃 Hinge	Hinge							
	Value								
	Write At The End 🗹 true	i							
	e tural feature which value should be represented for the configuration.								
	Close Back Forward	Help							

4. Set the created **CSV** export as a tagged value of the **executionListeners** tag definition of the Simulation Configuration. This step is important for the results to be written to file. See the CSV export configuration below:



Run the simulation and then stop it. The results will be written to the File Name specified in Step 3. See the parameters of CSV Export configuration below:

```
time (ms), clearance, d.width, a.width, b.width, c.width
0.0000, 0.4825, 34.3849, 1.9571, 1.9546, 29.9907
0.0000, 0.4258, 34.6383, 1.9683, 1.9703, 30.2738
0.0000, 1.0696, 34.3231, 1.9524, 1.9630, 29.3381
0.0000, 1.1681, 34.8364, 1.9729, 1.9542, 29.7412
0.0000, 0.5619, 34.7569, 1.9680, 1.9702, 30.2568
0.0000, 1.1400, 35.0440, 1.9520, 1.9714, 29.9807
0.0000, 1.2639, 34.6046, 1.9602, 1.9578, 29.4226
0.0000, 0.8996, 34.5504, 1.9554, 1.9690, 29.7265
0.0000, 1.3454, 34.5987, 1.9513, 1.9770, 29.3250
```

Support of Part properties

Not only can CSV export the values of properties with primitive types, e.g., Real, Integer, etc., but it can also export the types of the properties that are not primitive. Selected Part names in the **value** tag will be used as headers, and type names will be used as values of exported CSVs as shown in the figure below.

«CSVExport»		Α	В	С	D	E
Subtypes CSV	1	time(ms)	score	Main.wheel.brake.caliper	Main.wheel.brake.pad	Main.wheel.brake.rotor
<pre></pre>	2	0	-40.9345	Boss C12	Saphire 66	Rotus 26
	з	0	-42.5565	Cobra C3A	Saphire 66	Rotus 26
	4	0	-34.6917	Alphine K7	Saphire 66	Rotus 26
	5	0	-38.3437	Boss C12	Saphire 66	Rotus 275
	6	0	-39.863	Cobra C3A	Saphire 66	Rotus 275
	7	0	-32.496	Alphine K7	Saphire 66	Rotus 275
	8	0	-34.685	Boss C12	Saphire 66	Rotus 30
	9	0	-36.0593	Cobra C3A	Saphire 66	Rotus 30
Oinviteline Oceafia	10	0	-29.3952	Alphine K7	Saphire 66	Rotus 30
«Simulation Config» 🤤	× 11	0	-40.9345	Boss C12	Proto C9F	Rotus 26
«SimulationConfig» addControlPanel = false animationSpeed = 95 autoStart = true autoStartActiveObjects = true cloneReferences = false constraintFailureAsBreakpoint = false executionListeners = TexSubtypes CSV	12	0	-42.5565	Cobra C3A	Proto C9F	Rotus 26
	13	0	-34.6917	Alphine K7	Proto C9F	Rotus 26
	14	0	-38.3437	Boss C12	Proto C9F	Rotus 275
	15	0	-39.863	Cobra C3A	Proto C9F	Rotus 275
	16	0	-32.496	Alphine K7	Proto C9F	Rotus 275
	17	0	-34.685	Boss C12	Proto C9F	Rotus 30
executionTarget = BrakeTradeStudy_Subtype	s 18	0	-36.0593	Cobra C3A	Proto C9F	Rotus 30
fireValueChangeEvent = true	19	0	-29.3952	Alphine K7	Proto C9F	Rotus 30
InitializeReferences = false numberOfRuns = 1 recordTimestamp = false rememberFailureStatus = false resultLocation = 🗀 Subtypes TradeStudy runForksInParallel = true silent = false solveAfterInitialization = true startWebServer = false timeVariableName = "simtime" treatAllClassifiersAsActive = true	20	0	-40.9345	Boss C12	Titan P3OS	Rotus 26
	21	0	-42.5565	Cobra C3A	Titan P3OS	Rotus 26
	22	0	-34.6917	Alphine K7	Titan P3OS	Rotus 26
	23	0	-38.3437	Boss C12	Titan P3OS	Rotus 275
	24	0	-39.863	Cobra C3A	Titan P3OS	Rotus 275
	25	0	-32.496	Alphine K7	Titan P3OS	Rotus 275
	26	0	-34.685	Boss C12	Titan P3OS	Rotus 30
	27	0	-36.0593	Cobra C3A	Titan P3OS	Rotus 30

Selected Part names used as headers and type names used as values of exported CSVs.

Related pages

- Representing data from a CSV file in a line chart
 Exporting plots data to a CSV file