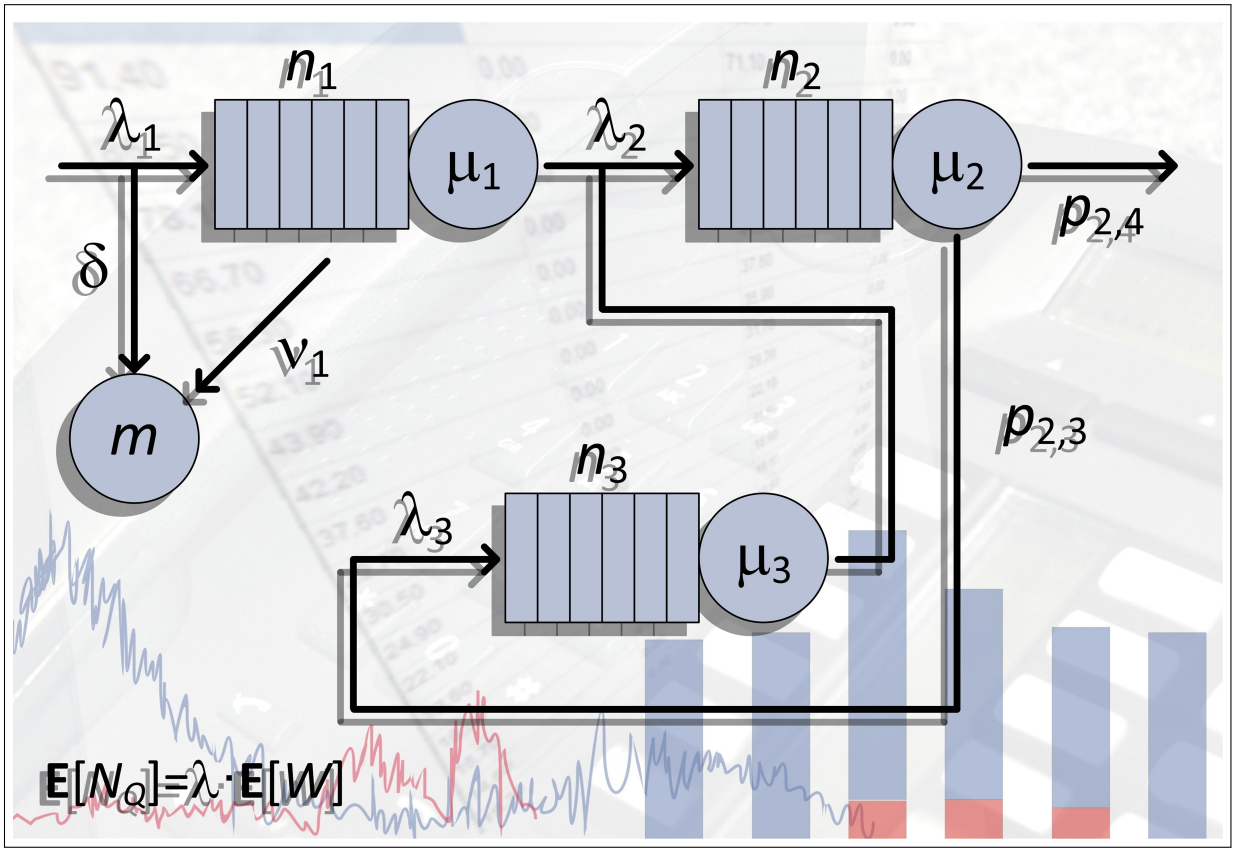
Command-line reference for Warteschlangensimulator

Alexander Herzog (alexander.herzog@tu-clausthal.de)



This reference refers to version 5.3.0 of Warteschlangensimulator.
Download address: https://github.com/A-Herzog/Warteschlangensimulator/.

Inhaltsverzeichnis

Kapitel 1
General information

Warteschlangensimulaktor is a Java program, i. e. a Java runtime environment is required for execution. If this is available, the simulator can be started with the following command:

```
java -jar Simulator.jar
```

If the file name of a model, statistics, parameter series or optimizer file is appended to this command line separated by a space, the file will be loaded immediately after the program starts.

Kapitel 2
List of commands

Alternatively, Warteschlangensimulator can be invoked with one of the commands described in the following sections. In this case no program window will be displayed, but all output is done directly to the command line. The calling scheme of the commands is as follows:

```
java -jar Simulator.jar Command OptionsForCommand
```

All commands described in the following can also be tried out directly via the graphical program interface using the command Ëxecute command-line command...“ in the Ëxtras"menu.

2.1 `Export`

Exports a model.

The command expects exactly two additional parameters:
1. Name of the input file
3. Name of the output file which will contain the exported model
The input file has to be existent, the output has to be non existent.

2.2 `Filter`

Filter statistic file.

The command expects exactly three additional parameters:
1. Name of the input statistic file
2. Name of the input filter configuration file
3. Name of the output file which will contain the filtered results
If the output file already exists, the new results will be added.
Otherwise the file will be created.

2.3 `Folder`

Simulates all models und parameter series in a folder.

The command expects the name of the folder to process as additional parameter.

2.4 FolderFilter

Filter all statistic files in a folder.

The command expects exactly three additional parameters:
1. Name of the input folder
2. Name of the input filter configuration file
3. Name of the output file which will contain the filtered results
If the output file already exists, the new results will be added.
Otherwise the file will be created.

2.5 Help

Shows this help.

This command expects one or no additional parameters.
If a command is enters as additional parameter, the help information for this command is displayed.
Otherwise a list with all available commands is displayed.

2.6 Interactive

Starts the interactive mode.

This command expects no additional parameters.

2.7 Optimizer

Starts a model optimization.

This command expects exactly two additional parameters:
1. Input model file
2. Optimizer configuration file
Both input files have to exist.

2.8 Outputtable

Processes an output table generated at a Save+Exit station.

This command expects two additional parameters:
1. Input table file
3. Output table file
The input file has to be existent, the output file has to be non existent.

2.9 ParameterSeriesVarianceAnalysis

Creates a parameter series configuration for a variance analysis

This command expects three or four additional parameters:
1. Input model file

2. Output parameter series file
3. Number of repetitions or the model
4. New value for the number of arrivals (optional)
The input file has to exist.
The output has to be non existent.

2.10 Parameterseries

Run a parameter series simulation.

This command expects exactly two additional parameters:
1. Input parameter series file
2. Output parameter series file
The input file has to exist.
The output has to be non existent.

2.11 ParameterseriesTable

Exports the results table of a parameter series.

This command expects exactly two additional parameters:
1. Input parameter series file
2. Output table file
The input file has to exist.
The output has to be non existent.

2.12 Report

Exports the whole report or a part of it.

This command expects exactly three additionall parameters:
1. „Inline„, „SingleFiles„, „List„, „Text„, „PDF„, „LaTeX„, „HTMLApp“ or a list entry, depending if
a) a html report with inline images,
b) a html teport with separate images,
c) an overview of all available individual documents,
d) a docx report,
e) a pdf report,
f) a LaTeX report
g) a html web app report or
h) an individual document is to be reported.
2. File name of the input file
3. File name of the output file

2.13 Server

Starts the program as simulation server.

This command expects 0 to 2 additional parameters.
If these parameters exist they have the following meaning:

1. Port to be used
2. Password for encrypted data transfer
If no parameters are specified, the simulator will ask for the port via the console.

2.14 ServerLimited

Starts the program as simulation server.

This command expects 0 to 2 additional parameters.
If these parameters exist they have the following meaning:
1. Port to be used
2. Password for encrypted data transfer
If no parameters are specified, the simulator will ask for the port via the console.
The server will limit the number of simultaneous requests depending on the number of available CPU cores.

2.15 ServerMQTT

Starts the program as MQTT-based simulation server.

This command expects two to four additional parameter:
the address of the MQTT broker, the MQTT topic optionally the name of a status information topic and optionally the user name and the password separated by „:„.

2.16 ServerMQTTFixed

Starts the program as MQTT-based simulation server for a fixed model.

This command three to five additional parameter:
the address of the MQTT broker, the MQTT topic, the file name of the model file to use, optionally the name of a status information topic and optionally the user name and the password separated by „:„.

2.17 ServerMQTTTest

Starts the program as MQTT-based echo test server.

This command expects two or three additional parameter:
the address of the MQTT broker, the MQTT topic and optionally the user name and the password separated by „:„.

2.18 ServerSocket

Starts the program as socket-based simulation server.

This command expects one or two additional parameters:
the port to be used and optionally a timeout value (in seconds) for individual simulations.

2.19 ServerWeb

Starts the program as web-based simulation server.

This command expects one or two additional parameter:
the port to be used and optionally the user name and the password separated by „:„.

2.20 ServerWebFixed

Starts the program as web-based simulation server for a fixed model.

This command expects two or three additional parameters:
the port to be used, the file name of the
model to be loaded and optionally the user name and the password separated by „:„.

2.21 Simulation

Starts a simulation run.

This command expects two or three additional parameters:
1. Input model file
2. Input data table file (optional)
3. Output statistic file
The input model file and the input data table (if specified) has to be existent, the output has to be non existent.

2.22 SimulationTimeout

Starts a simulation run (with timeout).

This command expects three or four additional parameters:
1. Input model file
2. Input data table file (optional)
3. Output statistic file
4. Timeout value (in seconds)
The input model file and the input data table (if specified) has to be existent, the output has to be non existent.