ETAS MODEL-SIMULATOR V2.1

User Guide

www.etas.com

Copyright

The data in this document may not be altered or amended without special notification from ETAS GmbH. ETAS GmbH undertakes no further obligation in relation to this document. The software described in it can only be used if the customer is in possession of a general license agreement or single license. Using and copying is only allowed in concurrence with the specifications stipulated in the contract.

Under no circumstances may any part of this document be copied, reproduced, transmitted, stored in a retrieval system or translated into another language without the express written permission of ETAS GmbH.

© Copyright 2023 ETAS GmbH, Stuttgart

The names and designations used in this document are trademarks or brands belonging to the respective owners.

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks.

MODEL-SIMULATOR V2.1 - User Guide R02 EN - 09.2023

Contents

1 Safety and Privacy Information ... 8
1.1 Intended Use ... 8
1.2 Target Group ... 8
1.3 Classification of Safety Messages ... 8
1.4 Privacy Notice ... 9
1.4.1 Technical and Organizational Measures ... 9
2 About MODEL-SIMULATOR ... 10
3 Basics of MODEL-SIMULATOR ... 12
3.1 Service Basics ... 12
3.2 Workflow ... 13
3.3 Projects ... 14
3.3.1 Simulation Projects ... 14
3.3.2 Simulations ... 15
3.3.2.1 Predicted Time and Predicted Costs of Simulations ... 16
3.3.2.2 Calculated Costs for Simulations ... 17
3.3.3 Test Execution Projects ... 18
3.3.4 Test Executions ... 18
3.3.4.1 Estimated Time and Estimated Costs of Test executions ... 19
3.3.4.2 Calculated Costs for Test executions ... 20
3.4 Campaigns ... 21
3.5 Models ... 21
3.5.1 Real Time Factor (RTF) ... 22
3.6 Compatibility of Campaigns and Models ... 24
3.7 Reports ... 26
3.8 Report Templates ... 26
3.9 Upload Files Overview and Supported File Formats (Artifacts) ... 27
3.9.1 Upload Files for MODEL-SIMULATOR ... 27
3.9.2 General Conventions for Uploaded Data ... 28
3.9.3 Required Data Structure of Subsystem campaigns ... 29
3.9.4 Required Data Structure of Virtual vehicle campaigns ... 29
3.9.5 Required Data Structure of Test Execution Campaigns ... 31
3.9.6 Required Data Structure of Models ... 34
3.9.7 Required Data Structure of Models (Use Case independent) ... 35
3.9.7.1 Files Specification of the simConfig Folder ... 37
3.9.7.2 Files Description of the simConfig Folder ... 38

3.9.8 Required Data Structure of Models (Test Execution Use Case) 40
3.9.9 Required Data Structure of Report Templates 40
3.10 Stimuli File Requirements (*.mf4) 40
3.11 Overview of Status Labels 41
3.11.1 Report Status in Simulation Projects 42
3.11.2 Simulation Status and Test Execution Status 42
3.11.3 Result Status in Test execution projects 43
3.11.4 Campaign, Model and Report Templates Status 44
3.11.5 Tooltips for Campaign Status 45
3.11.6 Tooltips for Model Status 47
3.11.7 Tooltips for Report Template Status 49
4 Getting Started 51
4.1 Check System Requirements 51
4.2 First Steps for Users 52
4.3 Check Model Data 54
4.4 Check Campaign Data 55
4.4.1 Data Structure for Subsystem Campaign 55
4.4.2 Data Structure for Virtual Vehicle Campaign 56
4.4.3 Data Structure for Test Execution Campaign 56
5 Working with MODEL-SIMULATOR 59
5.1 In a Nutshell 59
5.2 Process Steps 60
5.3 User Interface 61
5.4 Input Forms 62
5.5 Multi-Factor Authentication 63
5.5.1 Set up TOTP Authentication 63
5.5.2 Sign in via TOTP 64
5.6 Header 64
5.6.1 Change Password 65
5.7 Home 65
5.8 Simulation Projects 67
5.8.1 Open Simulation Projects Overview 67
5.8.2 Search Simulation Projects 68
5.8.3 Filter Simulation Projects Overview 68
5.8.4 Sort Simulation Projects Overview 68
5.8.5 Create a Simulation Project 69

5.8.6 Delete a Simulation Project 69
5.8.7 Delete Multiple Simulation Projects 69
5.8.8 Edit a Simulation Project 70
5.8.9 Open Simulation Project Details 70
5.8.10 Add Simulation Group(s) to a Simulation Project 70
5.8.11 Run Simulations in a Simulation Project 71
5.8.12 Stop Simulations in a Simulation Project 72
5.8.13 Delete a Simulation Group in a Simulation Project 73
5.8.14 Delete a Simulation Job in a Simulation Project 73
5.8.15 Delete a Simulation Run in a Simulation Project 74
5.8.16 Create a Report 74
5.8.17 Download a Report 75
5.8.18 Download Simulation Results 75
5.8.19 Open a Report 76
5.8.20 Delete a Report 76
5.9 Test Execution Projects 76
5.9.1 Open Test Execution Projects Overview 77
5.9.2 Search Test Execution Projects 77
5.9.3 Filter Test Execution Projects Overview 78
5.9.4 Sort Test Execution Projects Overview 78
5.9.5 Create a Test Execution Project 78
5.9.6 Delete a Test Execution Project 79
5.9.7 Delete Multiple Test Execution Projects 79
5.9.8 Edit a Test Execution Project 80
5.9.9 Open Test Execution Project Details 80
5.9.10 Add Test Execution Group(s) to a Test Execution Project 80
5.9.11 Run Test Executions in a Test Execution Project 81
5.9.12 Stop Test Executions in a Test Execution Project 82
5.9.13 Delete a Test Execution Group in a Test Execution Project 83
5.9.14 Delete a Test Execution Job in a Test Execution Project 83
5.9.15 Delete a Test Execution Run in a Test Execution Project 84
5.9.16 Download Test Execution Results 84
5.10 Campaigns 84
5.10.1 Status Labels for Campaigns 85
5.10.2 Required Preprocessing Steps for Campaigns 86
5.10.2.1 Preprocessing Steps for Virtual Vehicle Campaigns 87
5.10.2.2 Preprocessing Steps for Test Execution Campaigns 87
5.10.2.3 ECU-Test Support and Limitations for Test Execution Campaigns 87
5.10.3 MF4 File Creation Tools 89

5.10.4 Modify Existing MF4 File 90
5.10.5 Open Campaigns Overview 90
5.10.6 Search Campaigns 90
5.10.7 Filter Campaigns Overview 90
5.10.8 Sort Campaigns Overview 91
5.10.9 Create a Subsystem Campaign 91
5.10.10 Create a Virtual Vehicle Campaign 92
5.10.11 Create a Test Execution Campaign 92
5.10.12 Delete a Campaign 93
5.10.13 Delete Multiple Campaigns 93
5.10.14 Edit a Campaign 93
5.10.15 Open Campaign Details 94
5.10.16 Use Campaign(s) in a Simulation Project 94
5.10.17 Use Campaign(s) in a Test Execution Project 95
5.10.18 Download Stimuli Files of a Campaign 95
5.11 Models 96
5.11.1 Status Labels for Models 97
5.11.2 Use Cases 97
5.11.3 Required Preprocessing Steps for Models 98
5.11.4 Models in COSYM 99
5.11.5 Create a STI File 100
5.11.6 Create a SMF File 101
5.11.7 Open Models Overview 102
5.11.8 Search Models 102
5.11.9 Filter Models Overview 102
5.11.10 Sort Models Overview 103
5.11.11 Create a Model 103
5.11.12 Delete a Model 103
5.11.13 Delete multiple Models 104
5.11.14 Edit a Model 104
5.11.15 Edit Real Time Factor 104
5.11.16 Open Model Details 105
5.11.17 Use Model in a Project (Simulation Project or Test Execution Project) 105
5.11.18 Download Files of a Model 106
5.12 Report Templates 106
5.12.1 Status Labels for Report Templates 106
5.12.2 Open Report Templates Overview 107
5.12.3 Search Report Templates 107
5.12.4 Sort Report Templates Overview 108

5.12.5 Create a Report Template 108
5.12.6 Delete Report Template 108
5.12.7 Use or Change Report Template in a Simulation Project 109
5.12.8 Download Report Templates Files 109

6 Troubleshooting 111

7 Contact Information 112

Glossary 113

Figures 117

Tables 119

1 Safety and Privacy Information

In this chapter, you can find information about the intended use, the addressed target group, and information about safety and privacy related topics.

1.1 Intended Use

ETAS Cloud-Services is a cloud-based platform intended for virtualization purposes by providing a toolchain for continuous integration, test and validation of software in automotive electrical/electronic (e/e) systems.

With the MODEL-SIMULATOR it is possible to run simulations or automated test executions in the cloud, in order to significantly speed up this process in particular for large simulation and test tasks. As a result a report for simulation projects can be retrieved, and simulation or test execution results can be downloaded. The results of the simulation depend on the quality of the plant models and software models as well as on the choice of suitable excitation signals (input stimuli), quality and representativeness or tests procedures for the intended development task. The results should therefore be checked for suitability for subsequent investigations by the user.

ETAS Cloud-Services runs in a native cloud environment and is offered as a service (SaaS).

ETAS GmbH cannot be made liable for damage which is caused by incorrect use and not adhering to the safety information. See **Support** → **Safety advice** (Header).

1.2 Target Group

This product is directed at trained qualified personnel in the simulation and calibration sector of powertrain ECUs (e.g. calibration engineer, function developer or simulation model developer). Technical knowledge in simulation of vehicle systems and control unit engineering as well as pre-calibration or calibration of those is a prerequisite.

1.3 Classification of Safety Messages

Safety messages warn of dangers that can lead to personal injury or damage to property:

DANGER

DANGER indicates a hazardous situation that, if not avoided, will result in death or serious injury.

WARNING

WARNING indicates a hazardous situation that, if not avoided, could result in death or serious injury.

CAUTION

CAUTION indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.

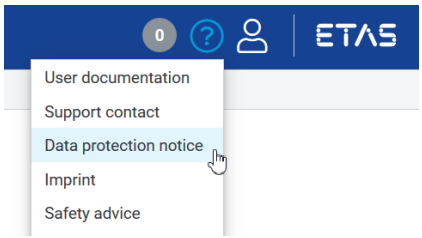
NOTICE

NOTICE indicates a situation that, if not avoided, could result in damage to property.

1.4 Privacy Notice

The purchaser of this software service is responsible for the legal conformity of processing the data in accordance with Article 4 No. 7 of the General Data Protection Regulation (GDPR). As the manufacturer, ETAS GmbH is not liable for any mishandling of this data.

For more information, please check the privacy policy in the support menu (top right corner of the header).



1.4.1 Technical and Organizational Measures

This service itself does not encrypt neither the personal data nor data categories that it records. Ensure that the data downloaded is secured by means of suitable technical or organizational measures in your own IT system.

2 About MODEL-SIMULATOR

ETAS Cloud-Services is a highly secured cloud-based platform, operated by ETAS GmbH. One of the offered services within this platform is a MODEL-SIMULATOR service, that enables massive parallelization of simulations with up to 1000 simulations in parallel. This massive parallelization of simulations allows many simulations within short time and thus increases software development efficiency for model based developments. Furthermore, massive parallelization of simulations enables statistical evaluation of systems.

The MODEL-SIMULATOR supports the simulation of models, that are generated with the ETAS product COSYM. Basically, these models can be open-loop or closed-loop models. Simulations can be triggered by combining a simulation model with a test scenario (campaign) within a project.

After successful simulation, the simulation outputs are aggregated in a report and can be downloaded for analysis. Besides a report, the MODEL-SIMULATOR provides simulation results for download.

Features at a Glance

- up to 1000 simulations in parallel
- allows simulation of complex systems like vehicle networks, consisting of virtual vehicle control units, vehicle buses and simulation models
- measurement of 1000 signals within every simulation run
- generating simulation results
- generating reports based on customized report layout
- highly secured cloud environment by fulfillment of information security standards according to ISO/IEC 27000 & 27001
- support of third party tool ECU-Test for test executions

Use Cases

- statistical evaluation of (complex) system models
- parameter studies with separate parametersets for simulations on simulation run level
- determination of the probability of failure against the development target
- time-efficient execution of (large) software testing tasks, e.g. regression tests in the DevOps work mode

Depenencies/Licenses

The uploaded artifacts should not contain:

- any reference to external libraries/dependencies
- any artifacts (e.g. FMUs/VECUs) which are needing additional licenses

> **Note**
>
> If you are using **VECU-BUILDER**, then use **Go License** for your artifacts.

3 Basics of MODEL-SIMULATOR

In this chapter, you find descriptions of the basic concepts of the MODEL-SIMULATOR to get an understanding of the service. It describes the typical workflow and its elements more detailed.

> **Note**
>
> If you want to start right away, check "Getting Started" on page 51 and "Working with MODEL-SIMULATOR" on page 59.

3.1 Service Basics

The MODEL-SIMULATOR enables the user to execute simulations in the cloud. Models and stimuli files are used as input, result files and reports are the output.

For this procedure the user has to create a simulation project or test execution project which consists of a simulation model and one or more campaigns. A campaign is integrated in a simulation group within the simulation project or in a test execution group within a test execution project. Beside simulation results, a report is generated after the simulation in a simulation project. The user can download the report, see Fig. 3-1.

The MODEL-SIMULATOR provides a model repository, where users can upload and manage simulation models. It also provides a Campaigns overview, where all campaigns can be uploaded and managed. In addition, users can upload a report template to create a customized report according to used signals.

Major elements of the MODEL-SIMULATOR are simulation models (models) and stimuli files (campaigns), which are brought together in a simulation project or in a test execution project.

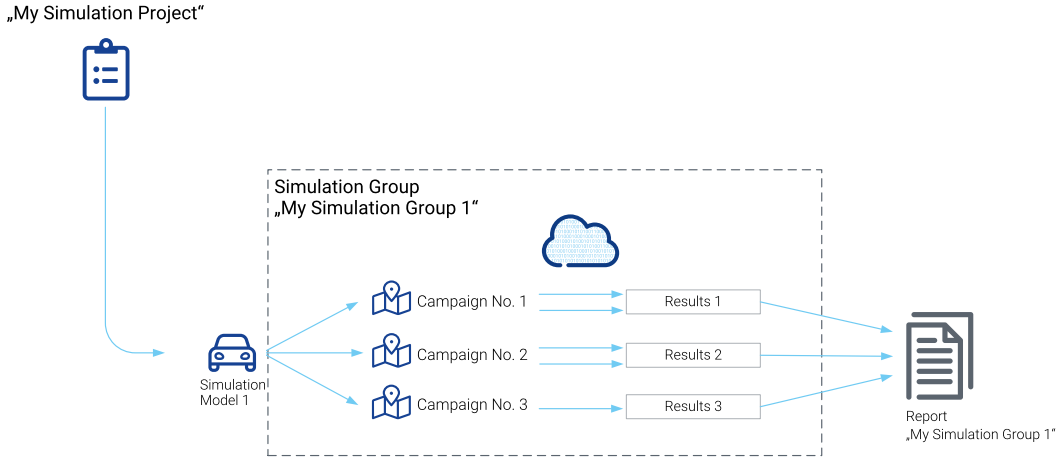


Fig. 3-1: Overview of MODEL-SIMULATOR elements and interaction (simulation project)

3.2 Workflow

The typical workflow in the MODEL-SIMULATOR is designed in the following steps, see in Fig. 3-2.

1. Create a model by uploading simulation model.
2. Create a campaign by uploading stimuli files for simulation or test execution.
3. Optional: Create a report template by uploading an EATB template (if not already available).
4. Create a simulation project or test execution project by selecting one model from the model repository and a default report template.
5. Add a simulation group to the simulation project or test execution group to test execution project by selecting campaigns.
6. Run the simulation in the simulation project or test execution in the test execution project.
7. Wait until the simulations or the test executions are completed.
8. Download the report (*.html) or simulation results as MDF files (*.mf4) or test execution results (*.zip).

Fig. 3-2: Overview workflow steps in the MODEL-SIMULATOR

For further information see:

- 5 "Working with MODEL-SIMULATOR" on page 59
- 5.11.3 "Required Preprocessing Steps for Models" on page 98
- 5.10.2 "Required Preprocessing Steps for Campaigns" on page 86

3.3 Projects

The Projects section in the MODEL-SIMULATOR provides the possibility to manage simulations and test executions. Within a project models and campaigns are combined to run simulations or test executions. In addition results and reports can be downloaded for further analysis or tests.

Two types of projects are available:

- 3.3.1 "Simulation Projects" below
- 3.3.3 "Test Execution Projects" on page 18

In simulation projects simulations can be run, in test execution projects automated test executions can be run.

3.3.1 Simulation Projects

Simulation projects are used to create and manage individual simulation projects. To assign one or multiple campaigns to a simulation project, a new simulation group has to be created. In a simulation project simulation models and stimuli files are brought together to realize the desired simulation task. The elements of a simulation project are visualized in Fig. 3-3.

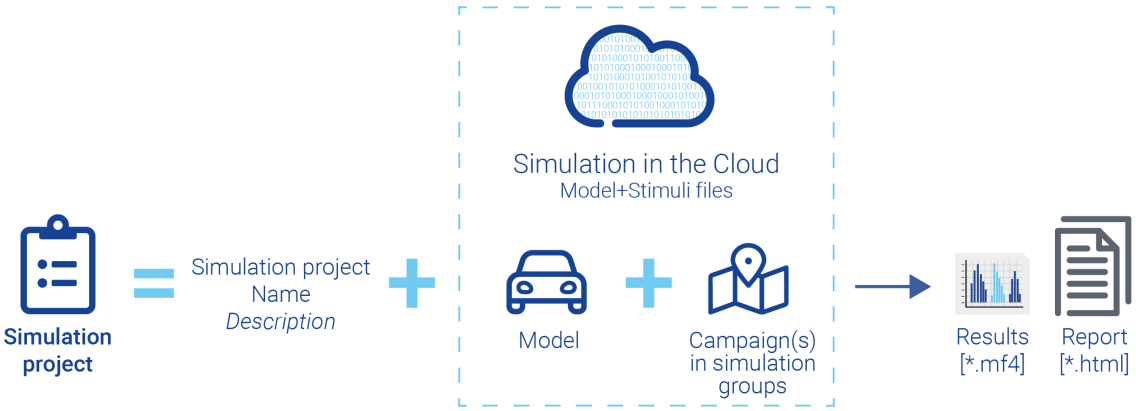


Fig. 3-3: Simulation project creation in MODEL-SIMULATOR and the containing data and outputs

Features Overview

The MODEL-SIMULATOR offers the following activities:

- Create, start or stop simulations of simulation project on different levels:
  - simulation groups
  - simulation jobs
  - simulation runs

  see 3.3.2 "Simulations" below
- Create or delete simulation group, simulation job and simulation run
- Check simulation status (see 3.11 "Overview of Status Labels" on page 41)
- Check predicted time and predicted costs
- Check simulation progress
- Download simulation results (MDF as *.mf4) on different levels:
  - simulation groups as *.zip
  - simulation jobs as *.zip
  - simulation runs as *.mf4
- Search or sort simulation projects
- Edit Project details (project name, description, default report template)
- Automatic report generation on simulation group level triggered by completed simulation groups
- Manual report generation on simulation group level
- Download reports on simulation group level

For instructions see chapter 5 "Working with MODEL-SIMULATOR" on page 59 and subchapter 5.9 "Test Execution Projects" on page 76.

3.3.2 Simulations

Simulations can be executed after combining one model with one or more campaigns within a simulation project. Simulations are an inherent part of the simulation project, which allows the user to run many simulations by assigning campaigns to the simulation project. To assign campaigns, simulation groups have to be created.

Simulations are represented by the simulation groups within a simulation project. The simulation can be based on one or multiple simulation groups, which are based on one ore multiple simulation jobs (campaigns). The simulation jobs itself are based on one or multiple simulation runs (trips). The structure is depicted in Fig. 3-4.

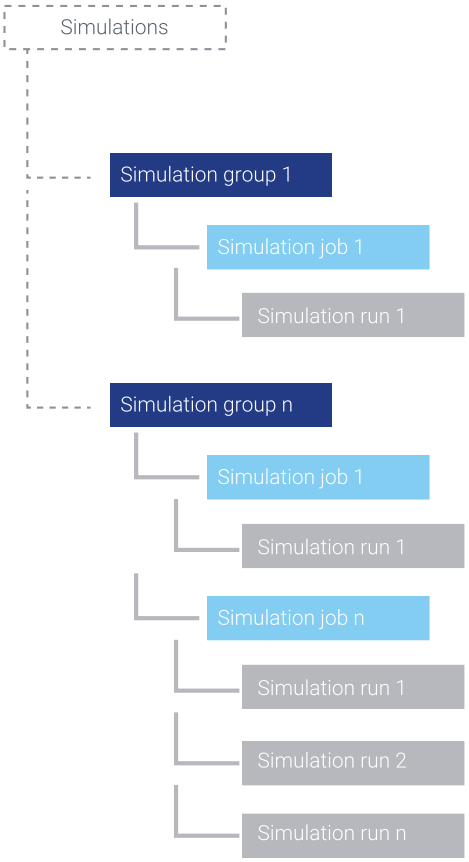


Fig. 3-4: Structure of simulation hierarchy

This simulation hierarchy with group, job and run level allows a flexible use. On every level simulations can be triggered and simulation results as MDF files (*.mf4) can be downloaded for analysis purposes.

The simulation group is the highest hierarchy level. By triggering a simulation on this level, all the simulations (simulation jobs and simulation runs within the simulation group) will be included into the simulation. Reports with a visualization of the data are generated on simulation group level. Starting a simulation on any level will lead to costs.
The report generation will be started once all the simulations within this simulation group are successfully finished.

3.3.2.1 Predicted Time and Predicted Costs of Simulations

The predicted simulation time differs from the real simulation time and has several influencing factors:

- Duration of stimuli file(s)
- Number of trips in simulation group
- Real time factor (RTF) of model

The predicted time is calculated by multiplying the duration of stimuli file with the RTF (see "Real Time Factor (RTF)" on page 22). The duration of MF4 files can be seen in the **Files** tab of a campaign.

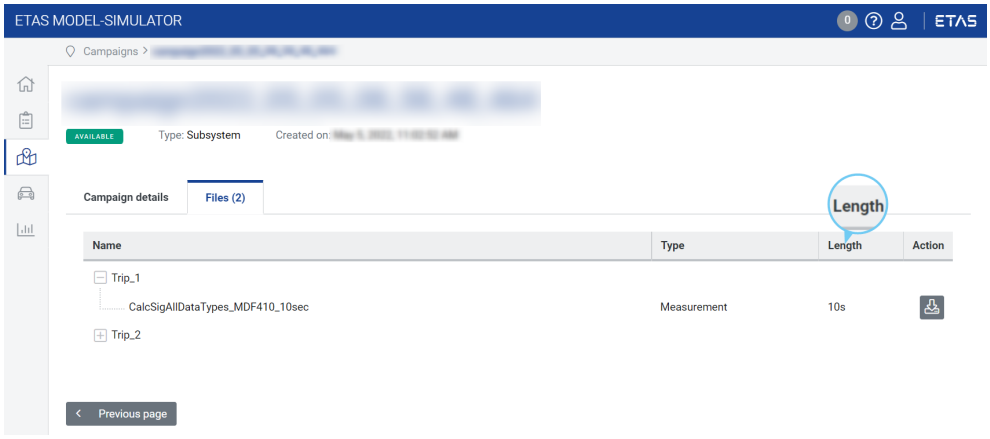


Fig. 3-5: Shown length in files tab of a campaign

Further, the required time for deployment for a single simulation node is added to the predicted time for every simulation run. This time is included in the shown value of predicted time.

For the cost prediction the predicted time is rounded up to a full hour. This is calculated on simulation run level and summed up on simulation job and simulation group level and gives you an indication about predicted costs.

> **Note**
>
> The predicted costs are an estimate based on the entered RTF.

Predicted costs and predicted time are shown in the simulations tab of a simulation project.

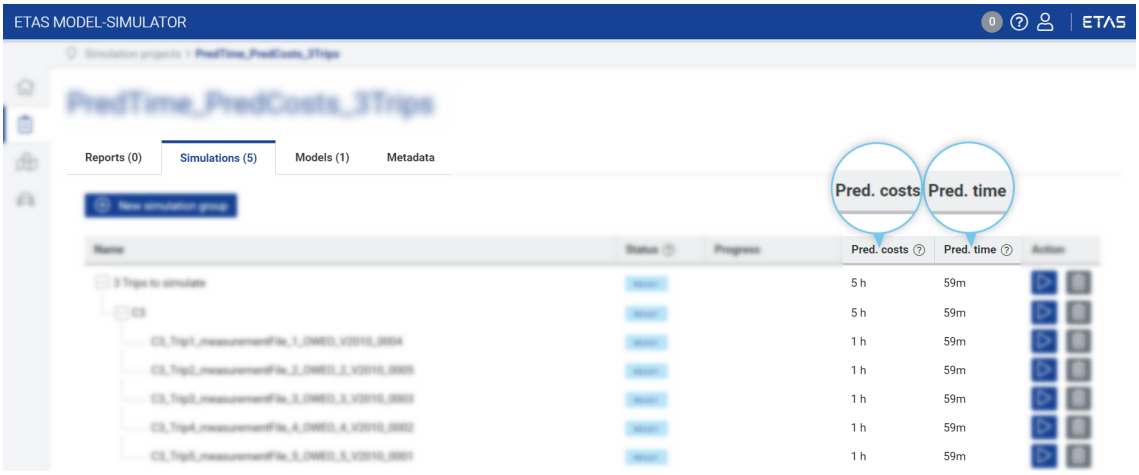


Fig. 3-6: Shown predicted costs and predicted time in simulations tab of a simulation project

3.3.2.2 Calculated Costs for Simulations

The MODEL-SIMULATOR costs are based on a pay-per-use concept. The currency is cloud computation hours.The costs will be calculated individually for each simulation run. The used amount of cloud computation hours will then be subtracted from the cloud computation hours contingent of your organization.

The displayed cloud computation contingent includes running simulations. Simulations can be run as long as your organization has enough contingent. The current status of cloud computation hours can be checked in the header: **My account > Cloud computation hours contingent**.

More information about accounting can be found in contractual documents.

3.3.3 Test Execution Projects

Test execution projects are used to create and manage individual automated software tests. To assign one or multiple campaigns to a test execution project, a new test execution group has to be created. In a test execution project simulation models and test configuration files are combined to run the test execution in the cloud. The automated test procedure is created with a test automation software (e.g. ECU-Test).

Features Overview

The MODEL-SIMULATOR offers the following activities:

- Create, start or stop test executions of test execution project on different levels:
  - test execution groups
  - test execution jobs
  - test execution runs

  see 3.3.4 "Test Executions" below
- Create or delete test execution group, test execution job and test execution run
- Check test execution (see 3.11 "Overview of Status Labels" on page 41)
- Download test execution results on different levels:
  - test execution groups as *.zip
  - test execution jobs as *.zip
  - test execution runs as *.zip
- Search or sort test execution projects
- Edit Project details (project name, description)

For instructions see chapter 5 "Working with MODEL-SIMULATOR" on page 59 and subchapter 5.9 "Test Execution Projects" on page 76.

3.3.4 Test Executions

During test executions, control mechanisms are applied to run the simulation so that the concept is not time-driven but event-driven. This means that events such as "engine on" lead to the next control step, e.g. "switch on ignition". Suitable controls must be implemented in the model.

Test executions can be run by combining one model with one or more test execution campaign within a test execution project. The executions are part of the test execution project, which allows the user to run many test executions by assigning campaigns to the test execution project. To assign campaigns, test execution groups have to be created.

Test executions are represented by the test execution groups within a test execution project. The test execution can be based on one or multiple test execution groups, which are based on one ore multiple test execution jobs (campaigns). The test execution jobs itself are based on one or multiple test execution runs (test units). The structure is depicted in Fig. 3-7.

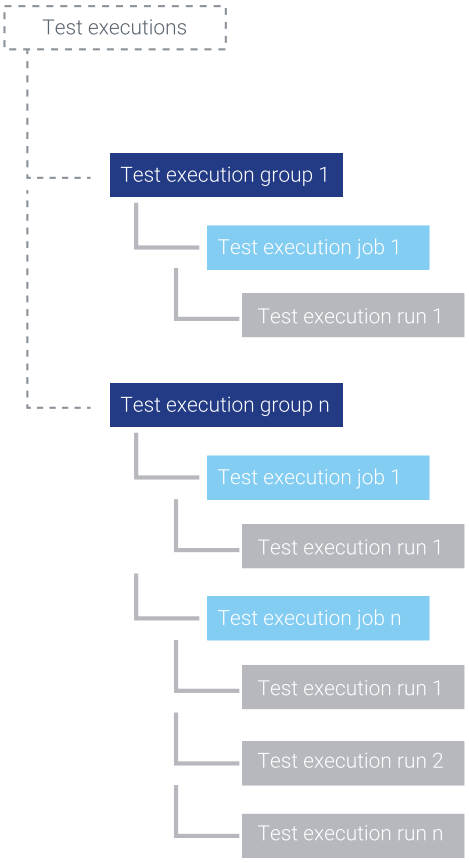


Fig. 3-7: Structure of simulation hierarchy

This test execution hierarchy with group, job and run level allows a flexible use. On every level test executions can be triggered and results can be downloaded.

The test execution group is the highest hierarchy level. By triggering a test execution on this level, all the test executions with all nested elements will be executed. Starting a test execution on any level will lead to costs.

3.3.4.1 Estimated Time and Estimated Costs of Test executions

Estimated time and estimated costs are shown in a pop-up right before starting a test execution. Due to the event-driven control of the simulation, the duration of the simulation is highly dependent on the behavior of the model. So, an

estimation of the duration is not possible in advance. Therefore, the user can only be informed which costs (cloud computation hours) will arise depending on the execution period. It is recommended to observe the test execution especially during the first execution in order to prevent unintentionally long test runs.

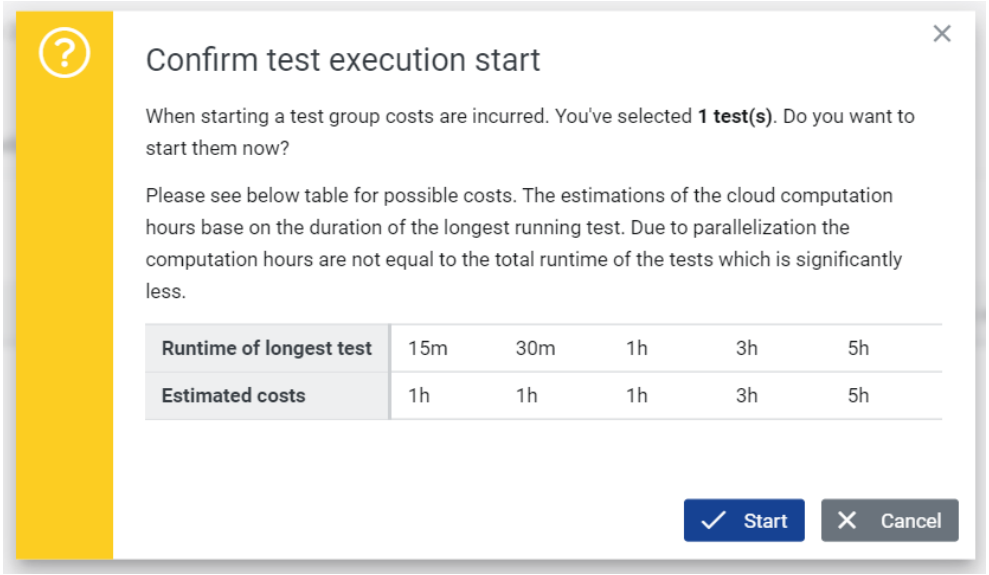


Fig. 3-8: Shown estimated time and estimated costs (tests tab of a test execution project)

3.3.4.2 Calculated Costs for Test executions

The MODEL-SIMULATOR costs are based on a pay-per-use concept. The currency is cloud computation hours. The costs will be calculated individually for each test execution run. The used amount of cloud computation hours will then be subtracted from the cloud computation hours contingent of your organization. Costs are charged in one hour steps.

The current status of cloud computation hours can be checked in the header: **My account › Cloud computation hours contingent**.

More information about accounting can be found in contractual documents.

3.4 Campaigns

Campaigns define the simulation task to be applied to the model. Campaigns are created and managed as separate elements so that they can be applied to different models. However, the compatibility between campaign and model must be ensured. Campaigns contain the information how a simulation should be executed.

The user can choose between 3 types.

- subsystem campaign: contains customer specific signals (*.mf4)
- virtual vehicle campaign: has to contain the signals vehicle target speed, gear and road slope (*.mf4). Parameters (*.cdfx) that are optionally defined will overwrite the parameters, which were defined within the model (CDFX file in the simConfig folder).
- test execution campaign: contains configuration of an automated test procedure created with a test automation software e.g. ECU-TEST (*.prj)

> **Note**
>
> The compatibility of campaign and model has to be ensured by the user.

See also 3.6 "Compatibility of Campaigns and Models" on page 24.

Features Overview

The Campaigns section offers the following activities:

- Create or delete campaigns
- Edit Campaign details (campaign name, description)
- Check and refresh status of campaigns
- Search or sort campaigns
- Download stimuli files in **Files** tab of a campaign (see 3.9 "Upload Files Overview and Supported File Formats (Artifacts)" on page 27):
  - measurement files as *.mf4
  - parameter files as *.cdfx (only for virtual vehicle campaign)
  - test units as *.prj (only for test execution campaign)

For instructions see chapter 5 "Working with MODEL-SIMULATOR" on page 59 and subchapter 5.10 "Campaigns" on page 84.

3.5 Models

A model is a mathematical representation of a physical system. It is used to calculate the resulting system behavior (model outputs) according to the fed excitation signals (input stimuli). The model contains the deployables of the simulation system (build target generated in COSYM V3.2) and some configuration files, like parameters or signal filter.

In the MODEL-SIMULATOR the user can upload these models for simulation and choose between two types: subsystem model and virtual vehicle model (see technical details in 3.9.6 "Required Data Structure of Models" on page 34). Whereas a virtual vehicle model has to contain the inputs vehicle target speed, gear and road slopeModels act as individual elements and can be assigned to different projects, where the model can be simulated by combining with campaigns. Once a model is uploaded in the model repository, it can be used for simulation of different campaigns.

> **Note**
>
> The compatibility of campaign and model has to be ensured by the user.

See also 3.6 "Compatibility of Campaigns and Models" on page 24.

3.5.1 Real Time Factor (RTF)

The RTF is a factor that describes how fast a simulation runs in comparison to real time. An initial value can be determined by running a simulation in COSYM and dividing the simulation time by the duration of stimuli file.

RTF = Duration of simulation/Duration of stimuli file

Example: If the simulation time (computing time on a machine) and the duration of stimuli file (driving time in the car) are equal and both take 1 hour, the RTF equals 1. If the simulation time takes more than the duration of stimuli file, the RTF is greater than 1.

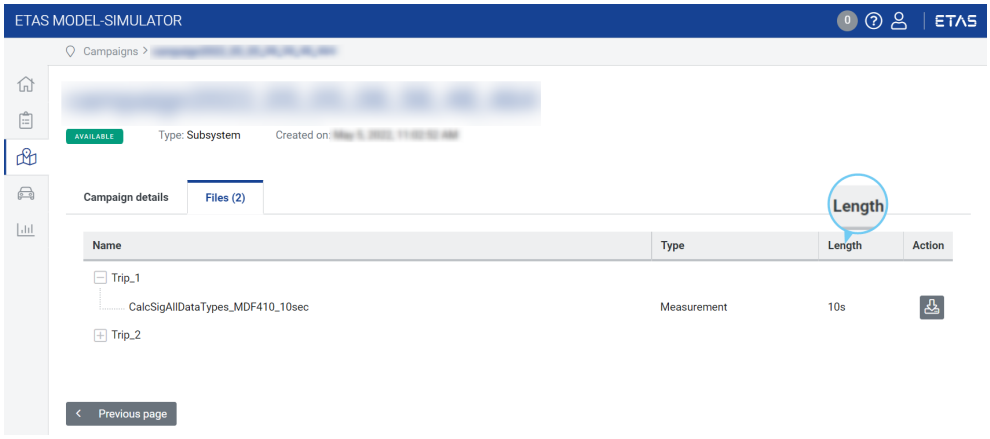
Duration of simulation ≙ Duration of stimuli file ⇒ RTF = 1

Duration of simulation > Duration of stimuli file ⇒ RTF > 1

Duration of simulation < Duration of stimuli file ⇒ RTF < 1

The RTF is entered during each model upload process to display the predicted simulation time. Therefore the determination is needed before uploading the simulation model to the MODEL-SIMULATOR. The RTF can be changed and adjusted afterwards in the model properties of a model. This change has an influence to **all** projects, wherever the model is used, but no influence on running simulations.

The RTF is used to calculate the predicted simulation time by multiplying it with the duration of stimuli file. The duration of MF4 files can be seen in the **Files** tab of a campaign.



Before a simulation starts, the predicted time is displayed in the simulations tab of a simulation project. This predicted time may differ from the actual required time, depending on how valid the RTF for all stimuli files is.

The RTF depends on the complexity of the simulation model, i.e. number of elements and stiffness of the differential equation system and the number and sampling frequency of the recorded channels. That means the lower the complexity, the lower the RTF will be.

A history of the RTF with statistics can be seen in the **Model properties** tab in every model. Based on run simulations a new RTF might be proposed which can be applied easily.

> **Note**
>
> The RTF value has no influence on the real simulation duration (charged hours) or the simulation speed. It only displays the predicted simulation time and predicted simulation costs based on the entered RTF.

Features Overview

The model repository offers the following activities:

- Create or delete models
- Edit Model details (model name, description)
- Edit Model properties (RTF)
- Check real time factor statistics
- Check and refresh status of models
- Search or sort models
- Download Files (see 3.9 "Upload Files Overview and Supported File Formats (Artifacts)" on page 27):
  - metadata as *.json
  - model binaries/executable as *.zip
  - parameter file as *.cdfx
  - signal mapping file as *.smf
  - datalogger configuration file as *.dlc4
  - interpolation file as *.sti

For instructions see chapter 5 "Working with MODEL-SIMULATOR" on page 59 and subchapter 5.11 "Models" on page 96.

3.6 Compatibility of Campaigns and Models

The combination of campaigns and models in a project has to be compatible. The compatibility is visualized in Fig. 3-9. Depending on model type and campaign type different inputs and signals have to be considered. This set up can be used for simulation or test execution use case, depending on the uploaded data.

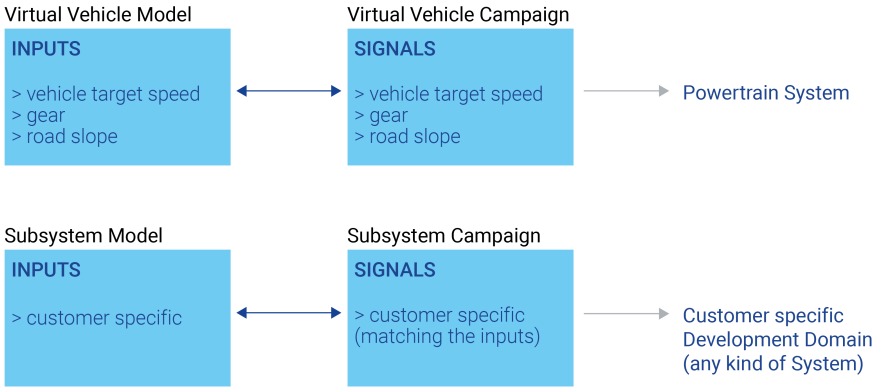


Fig. 3-9: Compatibility of inputs (models) and signals (campaigns)

The suitability of each model can be identified in the "Usecase" column of the model repository. Models with "Test" as usecase type can only be used in combination with test execution campaign in a test execution project. Models labeled

with "All" can be used either for test execution project and simulation project. Only the available models and campaigns are displayed in the corresponding project types during the project creation.

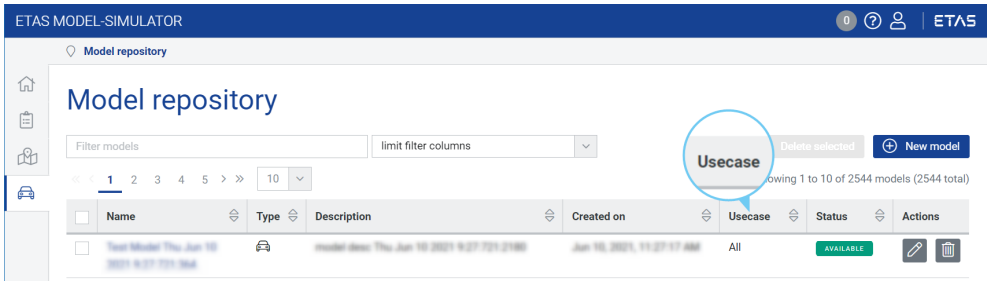


Fig. 3-10: Use case column in the model repository

3.7 Reports

Reports are used to visualize the simulation results of a simulation project. The Reports are created by another ETAS tool called EATB (ETAS Analytics Toolbox), which is integrated in the MODEL-SIMULATOR. After finishing a simulation, the results of a simulation group are collected to automatically generate an HTML report. That means all simulations belonging to a simulation group have to be completed, so that report creation starts. These automatically generated reports are named "Default report". Additionally, reports can be created manually with an own name and selected completed simulation groups. Reports must be downloaded in order to view them in the browser.
Reports are retained even if simulations are deleted, because simulation only has to be carried out successfully once. In addition, it is also possible to create reports manually. The reports are linked to templates on which they are based. That means you can create report templates according to your needs.

See also 4.1 "Check System Requirements" on page 51.
For more information, especially for usage and menu description, see separate EATB Report Guide or introduction video in the ETAS Download Center.

3.8 Report Templates

Report templates are used as a base for the report creation in a simulation project. They define the structure and design of the report. All templates are combined in a collection and can be managed as well as uploaded in a dedicated space. The easiest way to create a template is using EATB.

The report template is uploaded as a ZIP file containing at least one EATB (*.eatb) file for the different signals to be simulated and measured. The configuration file `config_signals.csv` or `config_signals.m` needs to be stored in the root folder.

3.9 Upload Files Overview and Supported File Formats (Artifacts)

In the MODEL-SIMULATOR different data has to be uploaded for campaigns and models. This chapter gives you an overview of all data types and the specifications.

3.9.1 Upload Files for MODEL-SIMULATOR

Object/Item	Description and Content	Format
Subsystem campaign	Stimuli file(s)	*.mf4 Version 4.1, ASAM Standard for MDF
Virtual vehicle campaign	Stimuli file(s) that contain vehicle target speed, gear and road slope. Zip file includes subordinate campaign folder with simulation run folders, each simulation run folder contains one MF4 file (mandatory) and one CDFX file (optional).	*.zip
Test execution campaign	ECU-Test files as a *.zip (without Report folder) that contains at least `Packages`, `Configurations` and `.workspace` folders.	*.zip

Object/Item	Description and Content	Format
Models	Virtual vehicle model: contains the description of the virtual vehicle with the inputs vehicle target speed, gear and road slope. Subsystem model: contains the description of a customer specific simulation model which in consequence has specific inputs. The ZIP file consists of `COSYMprj` folder with: › deployables as *.zip (mandatory) = cloud_deployable.zip of OS folder in COSYM project and `simConfig` folder with: – signal interpolation file as *.sti (mandatory) – datalogger configuration file as *.dlc4 (optional) – parameter file as *.cdfx (optional) – signal mapping file as *.smf (optional) **Note** A signal mapping file is only required if the labels in the CDFX file do not match the model's internal calibration variable names to ensure correct mapping. Otherwise, this could lead to wrong simulation results. Optionally the ZIP file can contain a `metadata` folder to submit custom data, e.g. reason for model creation, scope, etc. with: – metadata file as *.json For test execution projects only the deployables (cloud_deployable.zip) in `COSYMprj` folder as a *.zip is required.	*.zip

Tab. 3-1: Upload files overview for MODEL-SIMULATOR

See also 3.6 "Compatibility of Campaigns and Models" on page 24.

3.9.2 General Conventions for Uploaded Data

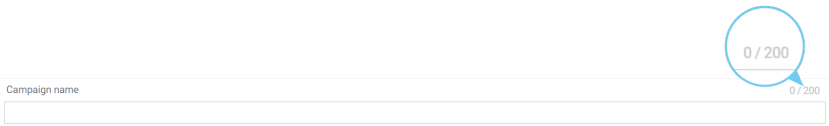
Following conventions have to be fulfilled for all uploaded data in MODEL-SIMULATOR:

- Maximum characters: 1024
- Can only start with any
  - lower case characters
  - upper case characters
  - numbers

    - special characters: _ .
- Can contain any
    - lower case characters
    - upper case characters
    - special characters like: ! - _ . ( )
- Not allowed:
    - other special characters like: '
    - white spaces

Due to character limitation of 140 characters for simulation group name, the name will be truncated. Each component can be up to 44 characters long. [Campaign_name]_[Folder_name]_[File_name]_[Sequence_number]
To ensure that the file name is not truncated, make sure that campaign name, files names of *.zip and *.mf4 are not longer than 44 characters.

Used characters are shown during creation at the right edge of the text input field.



3.9.3 Required Data Structure of Subsystem campaigns

In a subsystem campaign one or more stimuli files can be uploaded at once, each as an MDF file (*.mf4) considering the ASAM Standard for MDF.

See also 3.10 "Stimuli File Requirements (*.mf4)" on page 40. For conceptual background information see 5.10 "Campaigns" on page 84.

3.9.4 Required Data Structure of Virtual vehicle campaigns

For the virtual vehicle campaigns a dedicated data structure has to be fulfilled for a successful upload.

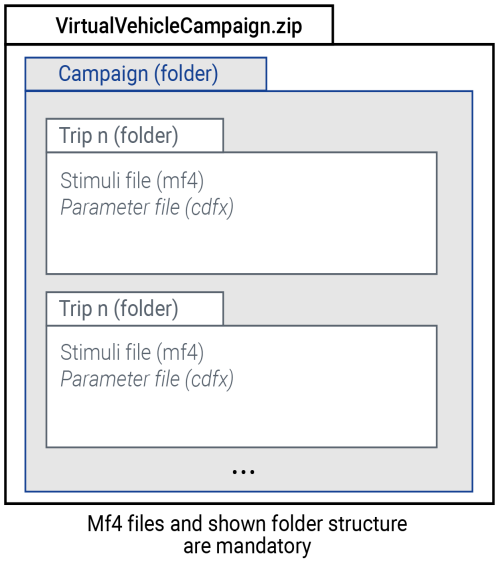


Fig. 3-11: Virtual vehicle campaign ZIP file composition (required structure)

A virtual vehicle campaign has to be uploaded as one ZIP file, e.g. `VirtualVehicleCampaign.zip` (see Fig. 3-12). This ZIP file must contain a superordinate folder containing one or multiple folders for each simulation run. Each of these folder must contain an MF4 file (ASAM Standard for MDF) and can contain a CDFX file (optionally), see in Fig. 3-13. The provided CDFX file(s) overwrite the values of the model CDFX file. If no CDFX file is provided, the parameters are constant and taken from the model.

There are no naming conventions, but you can lean on the examples in the screenshots. The used file names will be displayed in the Files tab of a campaign. See also 3.9.2 "General Conventions for Uploaded Data" on page 28.

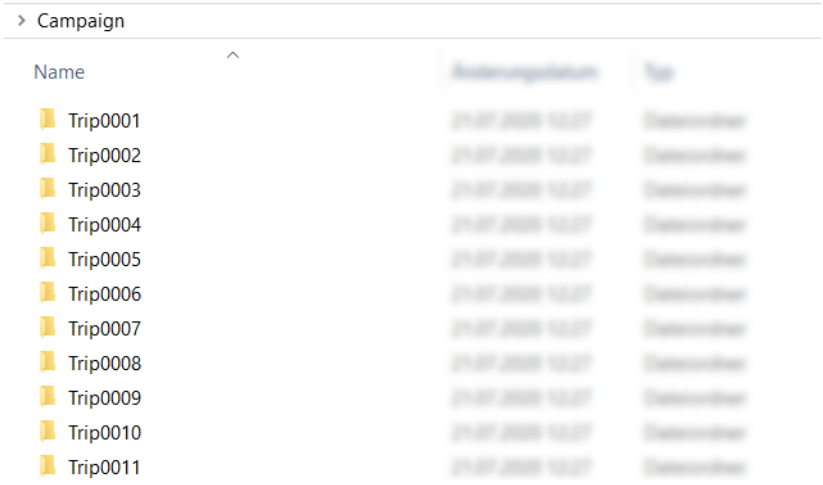


Fig. 3-12: Content of campaign folder in virtual vehicle campaign ZIP file

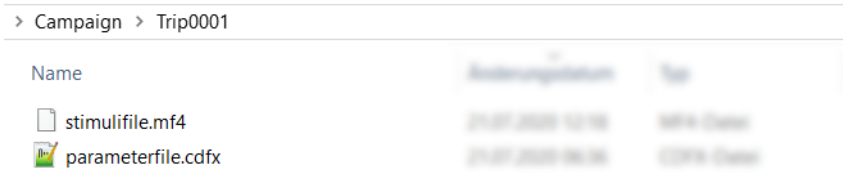


Fig. 3-13: Content of simulation run folder in virtual vehicle campaign

See also 3.10 "Stimuli File Requirements (*.mf4)" on page 40. For conceptual background information see 5.10 "Campaigns" on page 84.

3.9.5 Required Data Structure of Test Execution Campaigns

The ECU test tool now supports external dependencies provided as part of workspaces. An ECU test workspace that make use of a library feature can be uploaded via campaign services.

The use of the library feature is optional and both (with library feature and without library feature) campaign structures are supported.

For the test execution campaign a dedicated data structure has to be fulfilled for a successful upload.

> **Note**
>
> All the artifacts required to run the test cases must be packaged inside the workspace. The test bench configuration (.tbc) and test configuration (.tcf) files must contain only the relative paths to any artifacts they refer to within the workspace. If absolute paths are provided the test case execution in the cloud will fail.

Campaigns Without Use of Library Feature

The ECU test data has to be zipped (without Report folder) and uploaded for the test execution use case. At least the ZIP file must contain the packages, configurations and workspace folder of the ECU test project for campaigns without libraries.

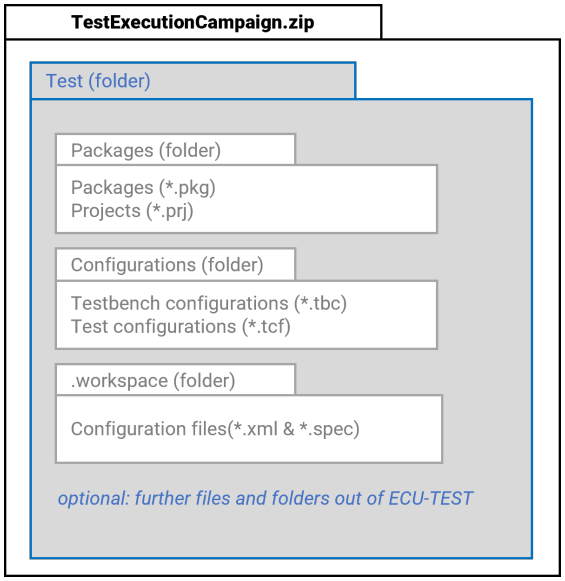


Fig. 3-14: Test execution campaign ZIP file composition (required structure) for campaigns without libraries

Campaigns With Use of Library Feature

For campaigns with libraries the following ZIP file composition has to be considered. ECU test data has to be zipped (without Report folder) and uploaded for the test execution use case. At least the ZIP file must contain a root folder, one or more library workspaces, a project folder with the packages, configurations and workspace folder of the . ECU test project for campaigns with libraries.

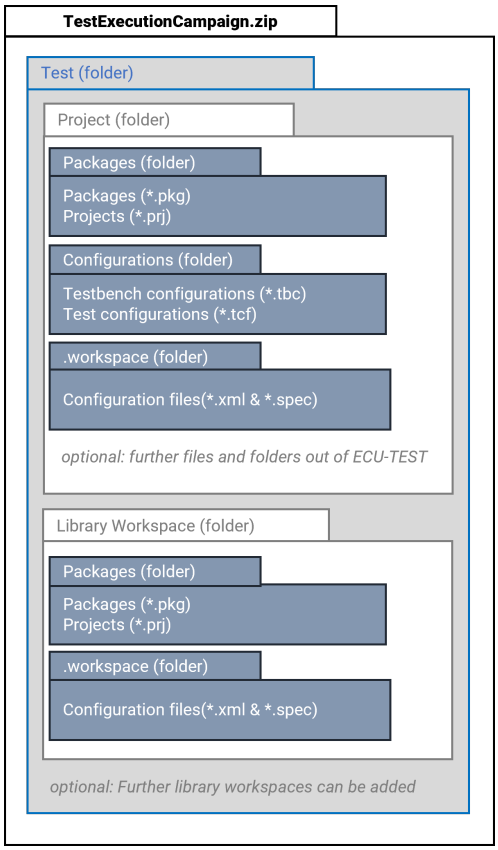


Fig. 3-15: Test execution campaign ZIP file composition (required structure) for campaigns with libraries

3.9.6 Required Data Structure of Models

Models are represented in ZIP files with a specific data container. A dedicated data structure has to be fulfilled for a successful upload. Depending on the use case, the following content is required:

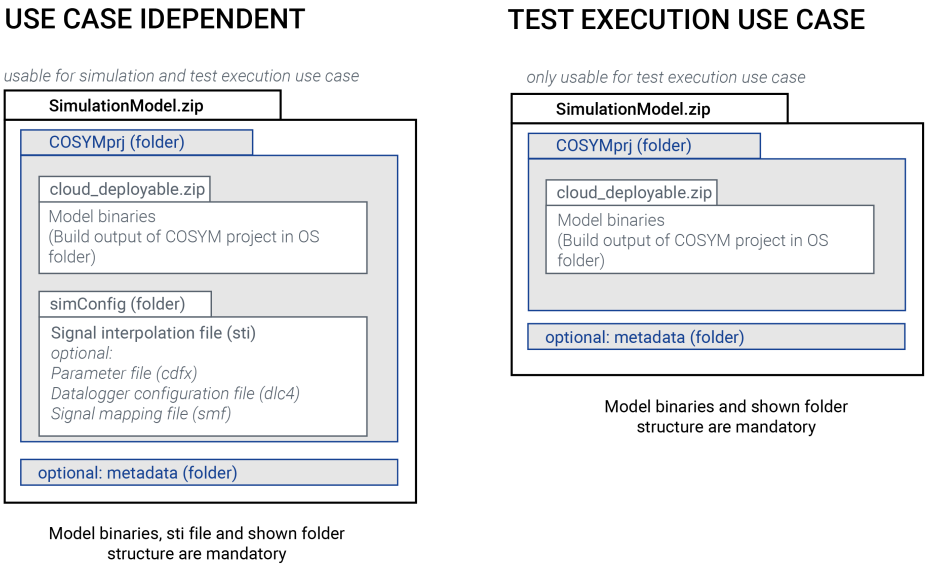


Fig. 3-16: Model ZIP file composition (required structure)

3.9.7 Required Data Structure of Models (Use Case independent)

A virtual vehicle model or subsystem model has to be uploaded as one ZIP file, e.g. `SimulationModel.zip` (see Fig. 3-16). This model ZIP file must contain a folder called `COSYMprj` and can contain a `metadata` folder (optionally). The `COSYMprj` folder must contain the deployables as a ZIP (build target generated with COSYM V3.2) and a configuration folder, which must be named `simConfig` and contain an STI file. The use of SMF file is optional. This construct can be used for simulation projects and test execution projects.

The data structure of subsystem model or virtual vehicle model is the same, the deployables are just composed differently in COSYM. In both cases COSYM creates a `cloud_deployable.zip`. You can enable cloud deployable creation via **View › Settings** in COSYM. For more details see topics "Settings" and "Build" in COSYM User Guide or press F1 in COSYM and navigate to *User Interface › Settings* and *System Build › Build*.

There are no naming conventions for the deployables, model ZIP file or configuration files, but you can lean on the example in the screenshot. The used file names will be displayed in the file information tab of a model.
See also 3.9.2 "General Conventions for Uploaded Data" on page 28.

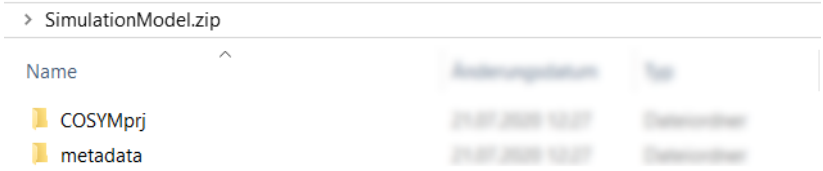


Fig. 3-17: Content of model ZIP file

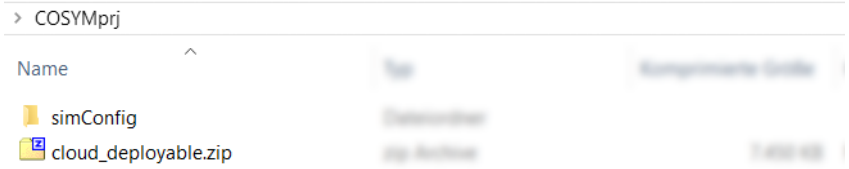


Fig. 3-18: Content of COSYMprj folder in model ZIP file (use case independent)

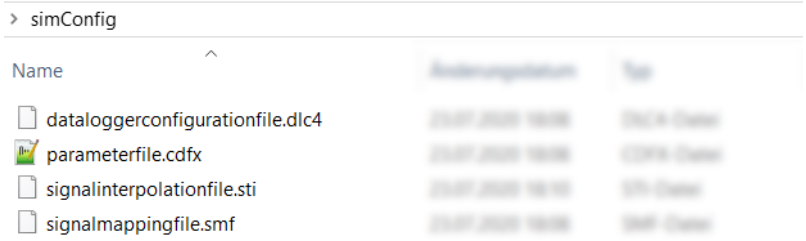


Fig. 3-19: Content of simConfig folder in COSYMprj folder

The `SimulationModel.zip` contains:

› `COSYMprj` folder with

- model binaries/executable as *.zip, e.g. `cloud_deployable.zip` (mandatory)
- configuration files in `simConfig` folder:
  - signal mapping file as *.smf (optional)

    > **Note**
    >
    > A signal mapping file is only required if the labels in the CDFX file do not match the model's internal calibration variable names to ensure correct mapping. Otherwise, this could lead to wrong simulation results.

  - parameter file as *.cdfx (optional)
  - datalogger configuration file as *.dlc4 (optional)
  - signal interpolation file as *.sti (mandatory)
- `metadata` folder

3.9.7.1 Files Specification of the simConfig Folder

Currently complex data types like maps, curves or arrays are supported, strings are not supported.

Object/Item	Description	Format	Creation Tool
Signal mapping file (optional)	Used for calibration parameter name mapping	*.smf Version 3.1 or 4.0 Only one SMF file, ETAS specific standard and format	Sut Mapping File Editor (usually comes with COSYM installation)

> **Note**
>
> A signal mapping file is only required if the labels in the CDFX file do not match the model's internal calibration variable names to ensure correct mapping. Otherwise, this could lead to wrong simulation results.

Parameter file (optional)	Global parameters of the model Just **scalar** elements are supported, **‹CATEGORY› type** of parameter must be therefore VALUE or BOOLEAN. Possible types: real64, real32, uint8, uint16, uint32, uint64, sint8, sint16, sint32, sint64. Example: `<CATEGORY>VALUE</CATEGORY>`	*.cdfx Version 2.0, ASAM standard for CDF Only one CDFX file containing Value or Boolean as Category type	COSYM, EE (Experiment Environment) or other ASAM CDF supported tool

Object/Item	Description	Format	Creation Tool
Note Model's internal variable names have to match with the parameter file (*.cdfx) within the model ZIP or in a virtual vehicle campaign you want to use for simulation. <SHORT-NAME> tag of CDFX (within <SW-INSTANCE> block) file has to use the same name as the label in the signal mapping file ("SUTMap Testlabel" in the SMF file).			
Datalogger configuration file (optional)	Signal selection for data recording	*.dlc4 Only one DLC4 file, ETAS specific file format	EE (Experiment Environment) → Datalogger tab
Interpolation file (mandatory)	Interpolation details variable values Just **linear** or **forward** interpolation is supported, **<Interpolation value>** must be therefore `eLinear` or `eForward`. Example: `<Interpolation xsi:type="InterpolationType" value="eLinear" />`	*.sti ASAM standard for STI Only one STI file	

Tab. 3-2: Model ZIP file: content of simConfig folder and specifications

Note

If a model is simulated with a virtual vehicle campaign, the provided CDFX file of the campaign overwrites the values of the model CDFX file.

See conceptual background information in 5.11 "Models" on page 96.

3.9.7.2 Files Description of the simConfig Folder

- `<SignalMappingFileName>.smf` (optional):

  Signal mapping file (*.smf) is used to provide proper mapping between labels inside the CDFX file with the model's internal calibration variable names, if they do not match. Otherwise this could lead to wrong simulation results.

- `<ParameterFileName>.cdfx` (optional):

  Global parameters of model used for initialization of model variables with predefined values.

- `<DataloggerConfigurationFileName>.dlc4` (optional):
- Signal selection for data recording, only the listed signals will be part of the simulation result instead of all. This reduces the data to be processed and speeds up the simulation. If no DLC4 file is provided, all signals will be

measured.
Data will be measured in 5 ms time steps. The step size cannot be changed.

- `<SignalInterpolationFileName>.sti` (mandatory):
  Global mapping of simulation model inputs to stimuli file signal names (e.g. the model uses input name **A** and the stimuli file use **X**, this must be mapped). Within this file you can address several mapping names coming out of different stimuli files. Time resolution in the stimuli file might be too low with respect to the simulation model requirements, therefore the input signals need to be interpolated to determine the stimuli file information within a finer timescale. Usable types of interpolation are: eLinear and eForward.

3.9.8 Required Data Structure of Models (Test Execution Use Case)

Regarding the test execution use case, only the deployables are required in a dedicated data structure. This model can then be only used for test execution projects. The metadata folder is an optional part of the ZIP file (see Fig. 3-16 in "Required Data Structure of Models" on page 34).

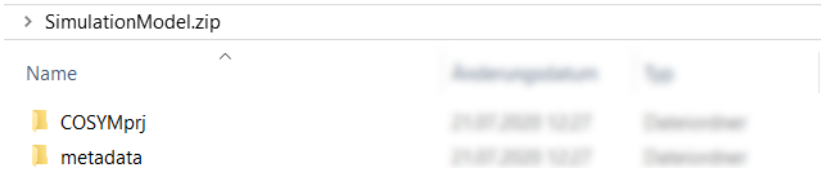


Fig. 3-20: Content of model ZIP file

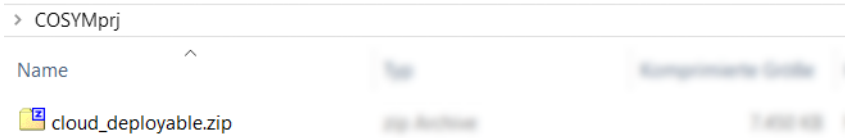
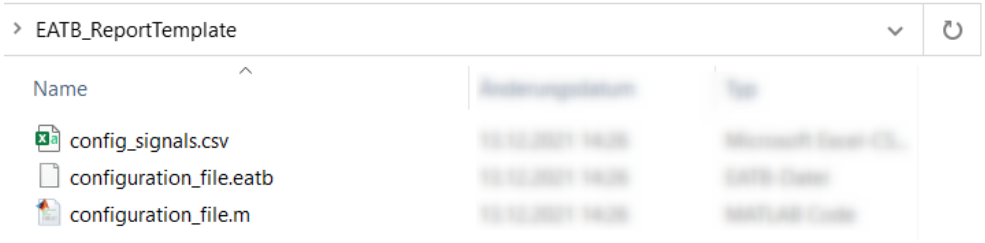


Fig. 3-21: Content of COSYMprj folder in model ZIP file (test execution use case)

3.9.9 Required Data Structure of Report Templates

The report template is uploaded as a ZIP file and can contain several EATB and M files for the different signals to be simulated and measured. The report template is created with the Configuration Creator in EATB. How to create a template is described in the separate User Guide of EATB, see Chapter "Configuration Creator". You can also watch an introduction video, see https://www.etas.-com/de/downloadcenter/39989.php.

For the upload, the ZIP file must contain at least one EATB file (*.eatb) and the `config_signals.csv` or `config_signals.m` provided in the root folder. The report template has to fit to the model outputs, otherwise the report will be empty.



3.10 Stimuli File Requirements (*.mf4)

MDF (*.mf4) is a binary format (ASAM standard) used for the stimulation of a model in the MODEL-SIMULATOR. The data is organized in data blocks. The channel group (CG) block contains channels (CN). One channel describes a recorded

signal including the encoding of the values in the records for the CG block.

The MDF format offers a wide range of possible applications. The following configuration is required for uploaded MF4 files to ensure a proper data processing:

- MDF 4.x format
- time stamps always start at 0 and are not negative
- only 1 channel group
- time unit is seconds
- sample rate is fixed

See more information: https://www.asam.net/standards/detail/mdf/

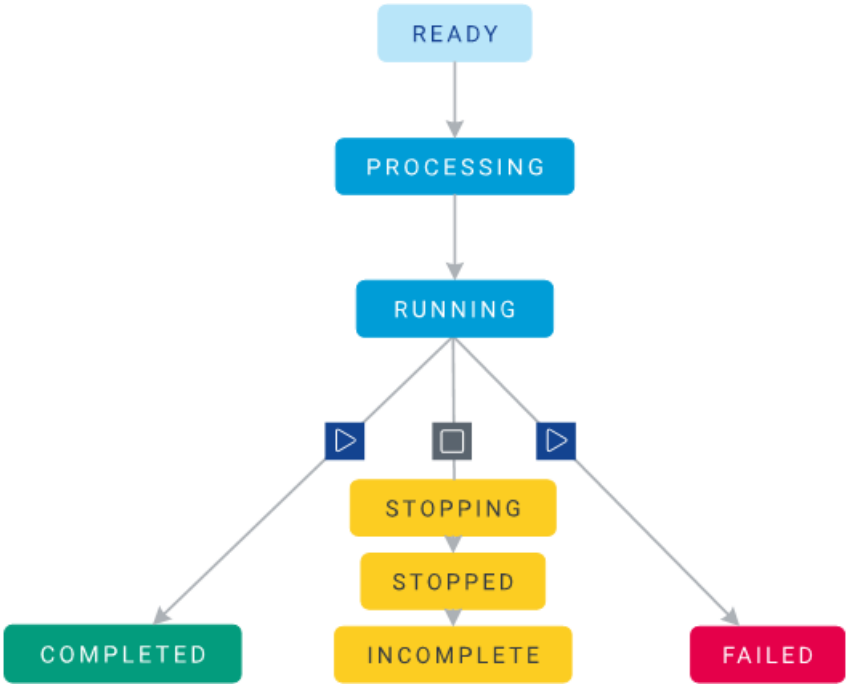
3.11 Overview of Status Labels

In the MODEL-SIMULATOR different status labels are defined and shown. The tables give an overview of all status labels and their meaning.

In general there is a color coding applied.

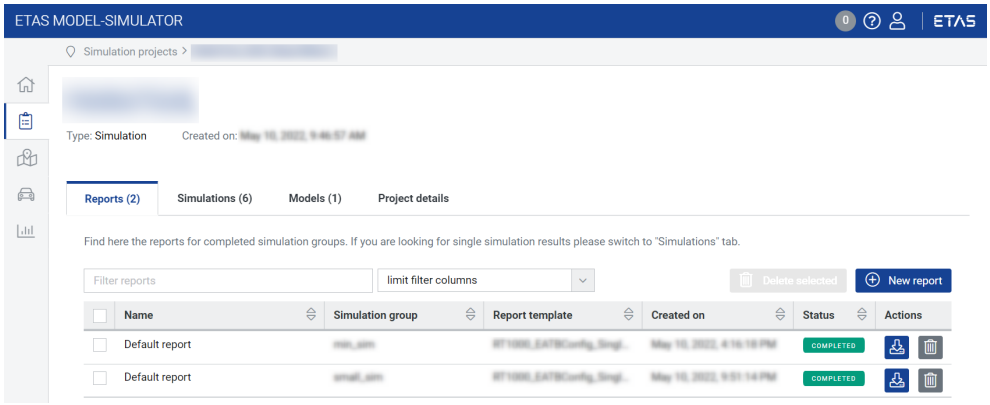
- light blue : ready, user action needed
- blue : in progress, process working
- green : finished, process completed
- yellow : stopped, process not completed
- red : not working, something went wrong, repeat process

There is a specific workflow followed for the status display after the simulation has been started. This sequence is shown in the following graphic.



3.11.1 Report Status in Simulation Projects

The report status can be found in each simulation project in the **Status** column of the **Reports** tab.

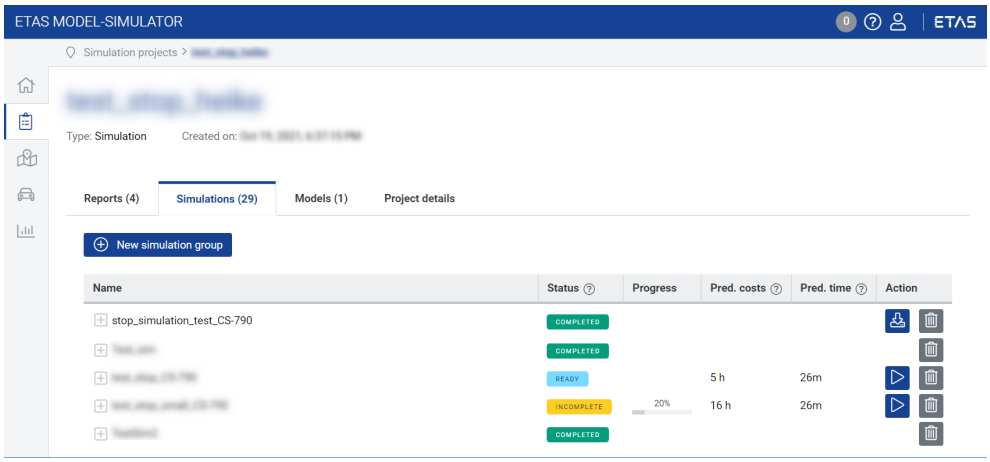


Labels	Description
PROCESSING	Report generation is triggered and in progress.
COMPLETED	Report is completed and ready for download. Report is named like simulation group and generated on simulation group level automatically.
FAILED	Report generation failed. Contact support for troubleshooting.

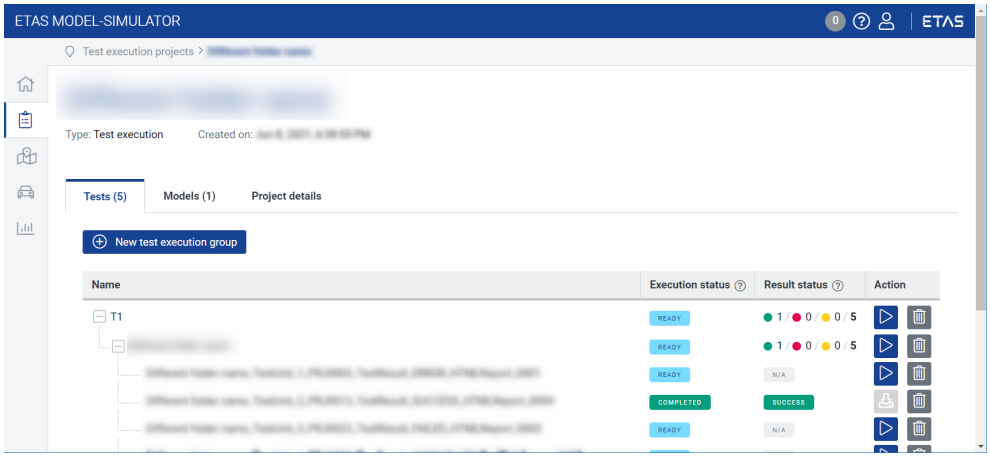
Tab. 3-3: Overview status labels for report status in simulation projects

3.11.2 Simulation Status and Test Execution Status

The simulation status can be found in each simulation project in the **Status** column of the **Simulations** tab.



The test execution status can be found in each test execution project in the **Execution status** column of the **Tests** tab.

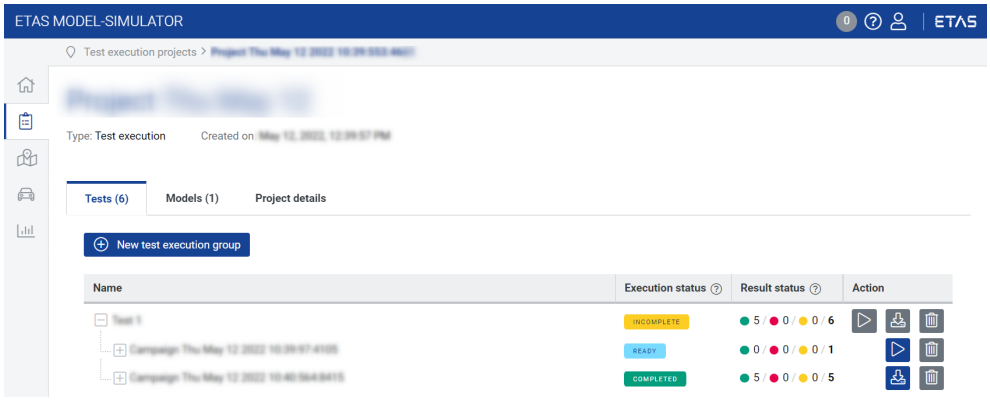


Labels	Description
READY	Simulation or test execution on group, job or run level is ready to use.
PROCESSING	Simulation or test execution is going to start. Setting up a simulation can take up some time, when a new simulation node has to be deployed.
RUNNING (only for simulation run and test execution run)	Simulation or test execution on run level is running and can be stopped by user in a certain time frame.
INCOMPLETE	At least one of the simulation run or test execution run was stopped by user, this leads to an incomplete job. Results of STOPPED simulations can be downloaded. It is recommended to avoid too many "incomplete" jobs in simulation projects, this can lead to issues for report generation. Incomplete label is also shown, if one or more runs in a job or group are complete but none is currently running.
COMPLETED	Simulation/test execution is completed.
FAILED	Simulation/test execution failed. Check files of campaign(s) and eventually start again. Possible other reason could be cloud computation hours limit is reached.

Tab. 3-4: Overview status labels for simulation status in simulation projects

3.11.3 Result Status in Test execution projects

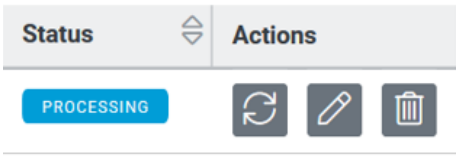
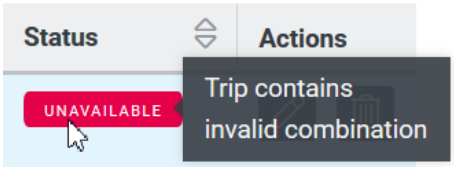
The result status can be found in each test execution project in the **Result status** column of the **Tests** tab.



Labels	Description
N/A	Test is not available due to not already started test execution. Execution status is READY.
FAILED	Test is failed due to not matching conditions.
INCONCLUSIVE	Trigger condition of trace analysis did not occur, no evaluation possible.
NONE	No comparison of actual and expected value in the project available.
SUCCESS	All test steps are passed with expected results.
ERROR	A tool error occurred.

3.11.4 Campaign, Model and Report Templates Status

The campaign, model or report template status can be found in the **Status** column of the overview pages or in each campaign, model or report template.

Labels	Description
PROCESSING	Campaign/model is created, but some process in the background is running (e.g. unzipping of files). Refresh and check the latest status, using the refresh button in Campaigns overview or model repository. 
AVAILABLE	Campaign/model is available and ready to use for simulation.
UNAVAILABLE	Campaign/model is created, but some background processing failed (e.g. unzipping). Campaign/model is not available and cannot be used for simulations. See more information in tooltips (hover over labels) inCampaigns overview or model repository.Check the reason for unavailability and eventually create a new campaign/model.  See also 3.11.5 "Tooltips for Campaign Status" below and 3.11.6 "Tooltips for Model Status" on page 47.

Tab. 3-5: Overview status labels for campaign/model status

3.11.5 Tooltips for Campaign Status

If a campaign is UNAVAILABLE, we provide detailed information in the tooltip for possible reasons. Hover over the label to see the details.

Tooltip	Description and Help
<No.> of <No.> routes failed due to internal error.	Campaign could not be created. Contact support for more details.
The campaign is unavailable because the .workspace/Packages/Configurations/ files were not found.	Test execution campaign must contain the following files: – .workspace folder with configuration files – Packages folder with PKG and PRJ files – Configurations folder with TBC and TCF files Check and see 3.9.5 "Required Data Structure of Test Execution Campaigns" on page 31.
The campaign is unavailable because the Configurations folder/s were not found.	
Stimuli length unavailable.	Check if MF4 requirements are fulfilled. See 3.10 "Stimuli File Requirements (*.mf4)" on page 40.
One or more files are present at a invalid path.	Data for virtual vehicle campaign must follow a dedicated structure. The folders within the ZIP file for a virtual vehicle campaign must only contain one MF4 file (Mandatory) and one CDFX file in each folder. Check and see 3.9.4 "Required Data Structure of Virtual vehicle campaigns" on page 29.
Invalid zip file: <filename>.zip	
Trip contains non accepted files other than MF4 and CDFX.	
Trip contains invalid combination.	
One or more empty trips are present.	
Artifacts contain invalid file formats.	
Folder exists already.	There is an internal error. Contact support to report the error, specifying the name.

Tooltip	Description and Help
Invalid characters in data context/Object name	There are some general conventions for the uploaded data. This is only checked after data upload. Ensure that the conventions are fulfilled and upload the data again. Check and see 3.9.2 "General Conventions for Uploaded Data" on page 28.
Artifacts contains invalid file formats.	Check that trips are stored in separate folders for virtual vehicle campaigns and each is provided with one MF4 file and one optionally CDFX file. See 3.9.4 "Required Data Structure of Virtual vehicle campaigns" on page 29.
One or more files are present at invalid path.	MF4 and optionally CDFX files have to be stored in folders for each trip. Check and see 3.9.4 "Required Data Structure of Virtual vehicle campaigns" on page 29.

3.11.6 Tooltips for Model Status

If a model is UNAVAILABLE, we provide detailed information in the tooltip for possible reasons. Hover over the label to see the details. The combination of single tooltips within one tooltip is possible, check separate descriptions.

Tooltip	Description and Help
Fail to unpack the uploaded model.	Data for virtual vehicle model & subsystem model must follow a dedicated structure. Use Winzip or 7-Zip for ZIP file creation. ZIP files created with MATLAB® or "send to compressed folder" feature of Windows are **not** working. See 3.9.6 "Required Data Structure of Models" on page 34 and "General Conventions for Uploaded Data" on page 28. Contact support for more details.
We could not find the simConfig folder. Please add one with mandatory config files to your model zip file.	Data for virtual vehicle model must follow a dedicated structure. Check and see 3.9.6 "Required Data Structure of Models" on page 34.
Mandatory zip file is missing.	The ZIP file for model upload must only contain one ZIP file. This ZIP file is the target build output of COSYM (cloud_deployable.zip). See 3.9.6 "Required Data Structure of Models" on page 34.
The ZIP file is present at invalid path.	The ZIP file for model upload must only contain one ZIP file (cloud_deployable.zip) stored in the COSYMprj folder. See 3.9.6 "Required Data Structure of Models" on page 34.
The SMF/CDFX file have an invalid file path structure.	SMF and CDFX file have to be stored in the simConfig folder. Check and see 3.9.6 "Required Data Structure of Models" on page 34.
Multiple SMF files are present.	Only one SMF file must be stored in the simConfig folder. Check and see 3.9.6 "Required Data Structure of Models" on page 34.
Multiple CDFX files are present.	Only one CDFX file must be stored in the simConfig folder. Check and see 3.9.6 "Required Data Structure of Models" on page 34.

Tooltip	Description and Help
Multiple DLC4 files are present.	Only one DLC4 file must be stored in the simConfig folder. Check and see 3.9.6 "Required Data Structure of Models" on page 34.
Multiple STI files are present	Only one STI file must be stored in the simConfig folder. Check and see 3.9.6 "Required Data Structure of Models" on page 34.
Multiple ZIP files are present.	The ZIP file for model upload must only contain one ZIP file. This ZIP file is the target build output of COSYM (cloud_deployable.zip). See 3.9.6 "Required Data Structure of Models" on page 34.
The files in the uploaded model have an invalid file structure.	Data for virtual vehicle model & subsystem model must follow a dedicated structure. Check and see 3.9.6 "Required Data Structure of Models" on page 34.

Tab. 3-6: Overview tooltips for model status

3.11.7 Tooltips for Report Template Status

If a report template is UNAVAILABLE, we provide detailed information in the tooltip for possible reasons. Hover over the label to see the details.

Tooltip	Description and Help
Path cannot exceed 1024 characters.	The total character length must not exceed 1024. See "General Conventions for Uploaded Data" on page 28.
Invalid characters in data context/Object name	There are some general conventions for the uploaded data. This is only checked after data upload. Ensure that the conventions are fulfilled and upload the data again. Check and see 3.9.2 "General Conventions for Uploaded Data" on page 28.
The report template zip should contain at least one EATB file.	Report template ZIP file must contain at least one EATB file (*.eatb or *.m). Check and see 3.9.9 "Required Data Structure of Report Templates" on page 40.

Tooltip	Description and Help
We found that either config_signals.csv or config_signals.m file or both are at an invalid path in the ZIP.	The configuration file `config_signals.csv` must to be stored in the root folder. Check and see 3.9.9 "Required Data Structure of Report Templates" on page 40.
Mandatory files are missing in the report template. Please upload a ZIP file with either config_signals.csv or config_signals.m file and at least one EATB file.	Report template ZIP file must contain at least one EATB file (*.eatb or *.m). The configuration file `config_signals.csv` must to be stored in the root folder. Check and see 3.9.9 "Required Data Structure of Report Templates" on page 40.
The ZIP file cannot be unpacked. The structure of the ZIP is invalid as one or more file(s) in the ZIP contains path as name.	Data for report template must follow a dedicated structure. Use Winzip or 7-Zip for ZIP file creation. ZIP files created with MATLAB® or "send to compressed folder" feature of Windows are **not** working. See 3.9.9 "Required Data Structure of Report Templates" on page 40 and "General Conventions for Uploaded Data" on page 28.

4 Getting Started

To get started with the MODEL-SIMULATOR some preparatory actions are required:

✓ User account created.

✓ Verification process completed.

✓ Authentication app is downloaded and configured (see "Set up TOTP Authentication" on page 63).

✓ System requirements are fulfilled (see "Check System Requirements" below).

✓ Data preparation are fulfilled (see "Check Campaign Data" on page 55 and "Check Model Data" on page 54).

4.1 Check System Requirements

Before starting with MODEL-SIMULATOR, be sure you meet the general system requirements.

- Required firewall settings for general use: HTTPS traffic enabled (port TCP 443)
- Required COSYM version and license for model preparation (simulation projects): Version V3.2 for simulation use case
- Required COSYM version and license for model preparation (test execution projects): Version V3.2 for test execution use case
- Required Visual Studio version and license for FMUs (FMI Version 2.0): 2013 or 2017

  Recommendation portable self-contained FMU without dependencies
- Required ECU-TEST version and license: V2023.2.3

  For ECU-TEST project (necessary for test execution campaign used in test execution project) without external references to projects or packages, see
- Required EATB version and license for template creation: V5.0
- Recommended browser version for reports at least:
  - Mozilla Firefox: v72.0.2 (64-bit)
  - Google Chrome: v84.0.4147.135(64-bit)
  - latest versions of the browser are recommended
- Recommended browser version for general use:
  - Mozilla Firefox: 91.8.0 (32-bit)
  - Google Chrome: 100.0.4896.127 (64-bit)
- Recommended screen resolution: 1920x1080 (Full HD), at least 1366x768 (Wide XGA)

See also supplied Release Notes for System Requirements.

> **Note**
>
> Check whether there are company-specific restrictions regarding upload size. This has to be enabled by your administrator if necessary.

4.2 First Steps for Users

This chapter gives you an overview and rough information of the first steps and how to start.

Activate your account

The activation link is sent to the registered e-mail address and valid for a limited time frame. Set your initial password as soon as possible. If your link got invalid, order a new one by contacting the support team or Admin User. Beside the MODEL-SIMULATOR account you get an account for the Customer Support Portal.

> **Note**
>
> You may check your spam e-mail folder as the registration e-mails sometimes got there.

Sign in with multi-factor authentication

With each sign in you will receive an authentication code as time-based one-time password (TOTP) to authenticate your user. See "Multi-Factor Authentication" on page 63.

Welcome to the dashboard

To successfully create a simulation project / test execution project (3) we recommend to start with your model upload (1) and your campaign creation (2).

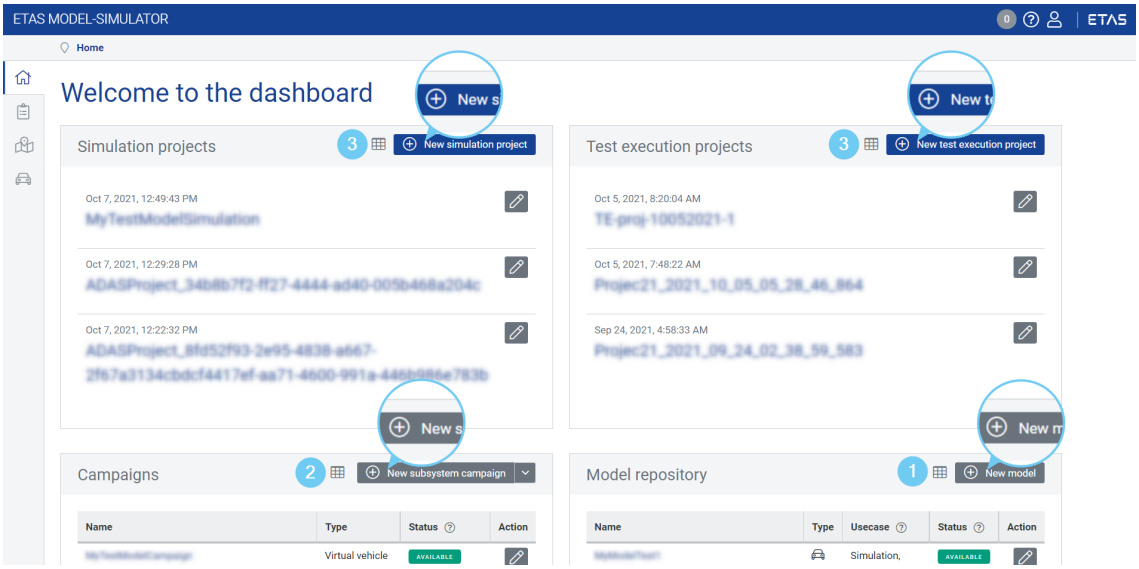


Fig. 4-1: First steps in the MODEL-SIMULATOR

Step 1: Upload your simulation model into the model repository

The model repository is a dedicated space to upload and manage all models.

You can either select a virtual vehicle or a subsystem.

Both model types require a ZIP file to be uploaded. The ZIP must include the deployables as a ZIP and a simConfig folder including an STI file at least – both in a folder called COSYMprj. See "Check Model Data" on the next page.
For instructions see chapter 5 "Working with MODEL-SIMULATOR" on page 59 and subchapter 5.11 "Models" on page 96.

> **Note**
>
> Only the deployables (`cloud_deployable.zip`) can be generated with COSYM. The overall ZIP file and the remaining files and the folder inside must be created and added manually.

You can enable cloud deployable creation via **View > Settings** in COSYM. For more details see topics "Settings" and "Build" in COSYM User Guide or press F1 in COSYM and navigate to *User Interface > Settings* and *System Build > Build*.

Step 2: Create a campaign and upload or generate your stimuli file(s)

Campaigns are used to manage simulation campaigns with individual scope.

You can either select a virtual vehicle campaign, subsystem campaign or test execution campaign according to your selected model.

A subsystem campaign requires one or more measurements files in MDF format (*.mf4). A virtual vehicle campaign must be uploaded as one ZIP file, with a subordinate folder containing one or multiple folders for each simulation run (trip). Each of these folders contains one stimuli file (*.mf4) and optionally one

parameter file (*.cdfx). For the test execution campaign you need a ZIP file with your ECU-Test project folder, without Report folder.
For instructions see chapter 5 "Working with MODEL-SIMULATOR" on page 59 and subchapter 5.10 "Campaigns" on page 84.

Step 3: Create a project and run simulations

Simulation projects or test execution projects are used to create and manage individual simulation or test execution tasks. Assign a model and one or multiple campaigns to the project.

To assign one or multiple campaigns to a project, a new simulation group/test execution group has to be created in the simulations/tests tab.
For instructions see chapter 5 "Working with MODEL-SIMULATOR" on page 59 and subchapter 5.8 "Simulation Projects" on page 67.

The outputs of simulations are reports (*.html) and simulation results (*.mf4) or test execution results (*.zip). They can be downloaded in the reports, simulations or tests tab.

4.3 Check Model Data

Simulation models used in MODEL-SIMULATOR have to be created with COSYM Software V3.2 for simulation projects, V3.2 for test execution projects). Other models cannot be processed by the application.

Data Structure

The application expects a dedicated folder structure.

The `SimulationModel.zip` contains:

› `COSYMprj` folder with

- model binaries/executable as *.zip, e.g. `cloud_deployable.zip` (mandatory)
- configuration files in `simConfig` folder:
  - signal mapping file as *.smf (optional)

    Note

    A signal mapping file is only required if the labels in the CDFX file do not match the model's internal calibration variable names to ensure correct mapping. Otherwise, this could lead to wrong simulation results.

  - parameter file as *.cdfx (optional)
  - datalogger configuration file as *.dlc4 (optional)
  - signal interpolation file as *.sti (mandatory)
- `metadata` folder

For test execution projects only the deployables (cloud_deployable.zip of COSYM) are required in the following composition.

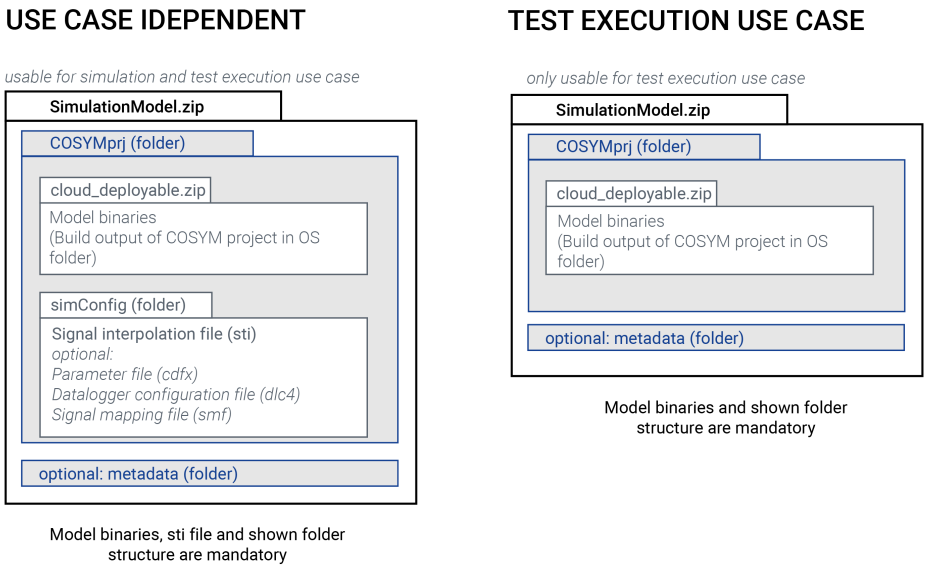


Fig. 4-2: Model ZIP file composition (required structure)

See also 3.9.6 "Required Data Structure of Models" on page 34 and 5.2 "Process Steps" on page 60.

All data has to be zipped in a file for upload. You can enable cloud deployable creation via **View › Settings** in COSYM. For more details see topics "Settings" and "Build" in COSYM User Guide or press F1 in COSYM and navigate to *User Interface › Settings* and *System Build › Build*.

> **Note**
>
> Only the deployables (`cloud_deployable.zip`) can be generated with COSYM. The overall ZIP file and the remaining files and the folder inside must be created and added manually.

See 5.11.3 "Required Preprocessing Steps for Models" on page 98.

4.4 Check Campaign Data

Stimuli files used in MODEL-SIMULATOR have to be created in an appropriate way. You must ensure the correct mapping of measuring channels from the time series and the model inputs by providing an SMF file (see 4.3 "Check Model Data" on the previous page). Deviating data cannot be processed by the application.

See also 3.6 "Compatibility of Campaigns and Models" on page 24.

4.4.1 Data Structure for Subsystem Campaign

The following dedicated requirements have to be fulfilled:

- *.mf4 Version (ASAM Standard for MDF)

After uploading the MF4 files are shown and downloadable in the **Files** tab of the campaign.

4.4.2 Data Structure for Virtual Vehicle Campaign

The following dedicated requirements have to be fulfilled:

- *.zip contains subordinate folder with one or multiple simulation run folders
- each folder with one *.mf4 (ASAM Standard for MDF) and optionally one *.cdfx file (ASAM Standard for CDF)

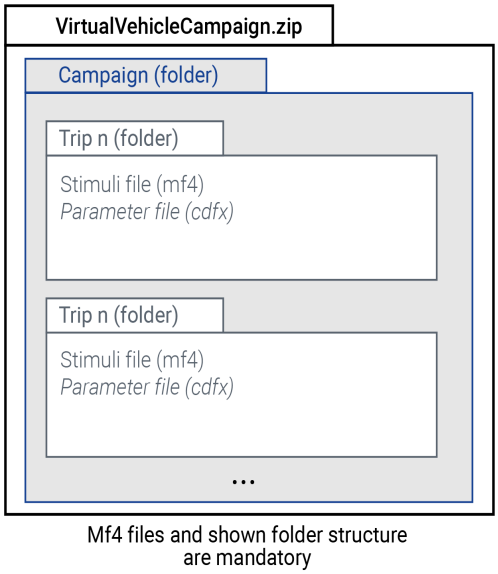


Fig. 4-3: Virtual vehicle campaign ZIP file composition (required structure)

See also 3.9.4 "Required Data Structure of Virtual vehicle campaigns" on page 29 and 3.10 "Stimuli File Requirements (*.mf4)" on page 40.

> **Note**
>
> It is necessary to ensure the compatibility of the simulation model and the used campaigns. See 3.6 "Compatibility of Campaigns and Models" on page 24

After upload the MF4 and CDFX files are shown and downloadable in the **Files** tab of the campaign.

4.4.3 Data Structure for Test Execution Campaign

The following dedicated requirements have to be fulfilled for campaigns without libraries:

- *.zip contains a folder (e.g. Test)
- test folder contains at least Packages, Configurations and .workspace folder of ECU-Test.

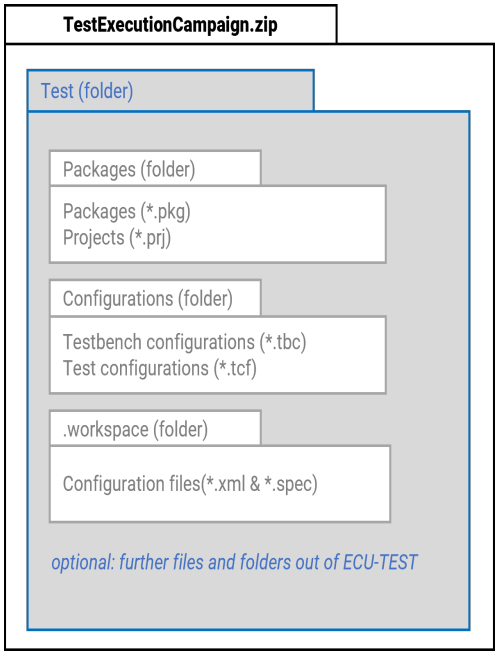


Fig. 4-4: Test execution campaign ZIP file composition for campaigns without libraries (required structure)

The following dedicated requirements have to be fulfilled for campaigns with libraries:

- *.zip contains a folder (e.g. Test)
- *zip contains one ore more library workspace folder
- *.zip contains the folder "Project"
- Project folder contains at least Packages, Configurations and .workspace folder of ECU test.

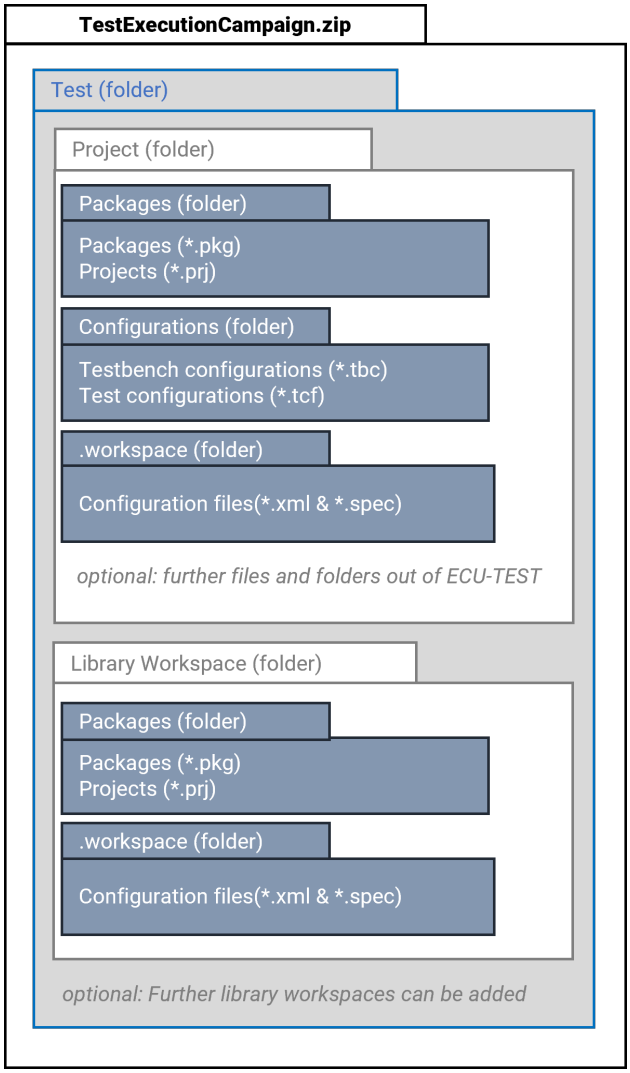


Fig. 4-5: Test execution campaign ZIP file composition (required structure) for campaigns with libraries

The ECU test project folder can be zipped (without Report folder) and uploaded. It is only important that the individual folders are embedded in a parent folder and that this folder is zipped. In the **Files** tab of the campaign only PRJ files are shown and downloadable.

5 Working with MODEL-SIMULATOR

The *Working with* section provides information with focus on step-by-step instructions as how-tos. In addition, the subchapters also contain descriptions at the beginning.

The content type can be identified by the heading:

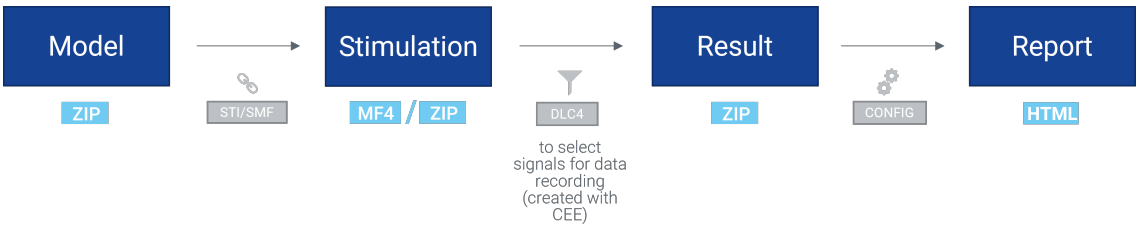
- Simulation projects (Heading *without verb*)
  → like "What is ..?" = description
- Create Simulation project (Heading *with verb*)
  → like "How to ..?" = instruction

Conceptual background of the MODEL-SIMULATOR is described in 3 "Basics of MODEL-SIMULATOR" on page 12.

5.1 In a Nutshell

For the work with the MODEL-SIMULATOR you need models and campaigns (stimulation). With these 2 elements you can run simulations or test executions in the cloud.

To get a rough holistic picture and the corresponding information, check the following graphic and the listed cross-references to the corresponding topics.



For some procedures within the MODEL-SIMULATOR preprocessing steps with other tools are necessary.

- Models: A dedicated structure for ZIP file (including deployables, SMF, CDFX, DLC4 and STI files) to be uploaded is needed, see "Check Model Data" on page 54 and "Required Preprocessing Steps for Models" on page 98.
- STI/SMF
  - STI: The STI file is used for signal mapping between stimuli (MF4) and models. It also supports interpolation of input signals. This file is part of the model ZIP file and is mandatory. See "Create a STI File" on page 100.
  - SMF: The SMF file is used to map inputs of the model with the signals of the campaign. This file is part of the model ZIP file and can be created with the Sut Mapping File (SMF) editor. SMF ile is optional, see "Create a SMF File" on page 101.

> **Note**
>
> A signal mapping file is only required if the labels in the CDFX file do not match the model's internal calibration variable names to ensure correct mapping. Otherwise, this could lead to wrong simulation results.

- Stimulation
  - Subsystem campaign: Several MF4 files can be uploaded, no preprocessing steps required.
  - Virtual vehicle campaign: A dedicated structure for ZIP file to be uploaded is needed, see "Data Structure for Virtual Vehicle Campaign" on page 56 and 5.10.2 "Required Preprocessing Steps for Campaigns" on page 86.
  - Test execution campaign: A dedicated structure for ZIP file to be uploaded is needed, see "Data Structure for Test Execution Campaign" on page 56 and 5.10.2 "Required Preprocessing Steps for Campaigns" on page 86.
- DLC4: The DLC4 file is part of the model ZIP file and defines which signals should be recorded and therefore be part of the results and report.
- Result: The results are part of the simulation or test execution and can be downloaded in the simulations or tests tab of a project.
- CONFIG: The ZIP file defines layout and structure of the report. This configuration can be created with EATB. The ZIP file has to be uploaded in the report template section.
- Report: Once all simulations of a simulation group are finished, a report will be generated automatically. The report can be downloaded in the reports tab of a simulation project. A report generation can also be started manually.

In the following part of the User Guide the focus is on instructions. If you are looking for background information, check 3 "Basics of MODEL-SIMULATOR" on page 12.

5.2 Process Steps

The typical overall workflow is designed in the following steps.

1. Preprocessing for models (not covered by the MODEL-SIMULATOR):
   - Integrate models in COSYM V3.2 on the PC.
   - Build the deployables (Target = SiL_target) in COSYM V3.2 on the PC. (You can enable cloud deployable creation via **View › Settings** in COSYM. For more details see topics "Settings" and "Build" in COSYM User Guide or press F1 in COSYM and navigate to *User Interface › Settings* and *System Build › Build*.)

- Mandatory: Create the interpolation file (*.sti) with a text editor and the sample file format, see "Create a STI File" on page 100.
- Optional: Create the signal mapping file (*.smf) with Sut Mapping File Editor (tool usually comes with COSYM), see "Create a SMF File" on page 101.
- Optional: Create the parameterization file (*.cdfx) with COSYM, EE (Experiment Environment) or other ASAM CDF supported tool.
- Optional: Create the datalogger configuration file (*.dlc4) with EE (Experiment Environment)→ Datalogger tab.
- Configure the model ZIP file on the PC and check all files (see 3.9.6 "Required Data Structure of Models" on page 34).
- Integrate models in COSYM V3.2 on the PC.

2. Sign in to the MODEL-SIMULATOR.
3. Create model by uploading simulation model (see 5.11.11 "Create a Model" on page 103).
4. Optional: Preprocess virtual vehicle campaign (not covered by MODEL-SIMULATOR)
   see 4.4 "Check Campaign Data" on page 55 or 3.9 "Upload Files Overview and Supported File Formats (Artifacts)" on page 27.
5. Create campaign by uploading stimuli file(s) or campaign ZIP file for simulation (see 5.10.9 "Create a Subsystem Campaign" on page 91, 5.10.10 "Create a Virtual Vehicle Campaign" on page 92 or "Create a Test Execution Campaign" on page 92).
6. Optional: Create a report template by uploading an EATB template (if not already available).
7. Create simulation project or test execution project(see 5.8.5 "Create a Simulation Project" on page 69 and "Create a Test Execution Project" on page 78).
8. Run simulation or test execution in the project (see 5.8.11 "Run Simulations in a Simulation Project" on page 71 and 5.9.11 "Run Test Executions in a Test Execution Project" on page 81).
9. Download report (see 5.8.17 "Download a Report" on page 75) or simulation results (see 5.8.18 "Download Simulation Results" on page 75) / test execution results (see "Download Test Execution Results" on page 84).

5.3 User Interface

The user interface is divided into 4 areas:

- **Header (1)**: gives you direct access to, among others, home screen, notifications of current processes or user documentation.
- **Breadcrumb (2)**: shows where you are and allows you to use for navigation.

- **Navigation bar (3)**: provides you access to home (on page 65), Projects (on page 67), Campaigns (on page 84) model repository (on page 96) and Report templates (on page 106).
- **Interaction area (4)**: Depending on the selection in the navigation bar, different content is displayed and different features are available.

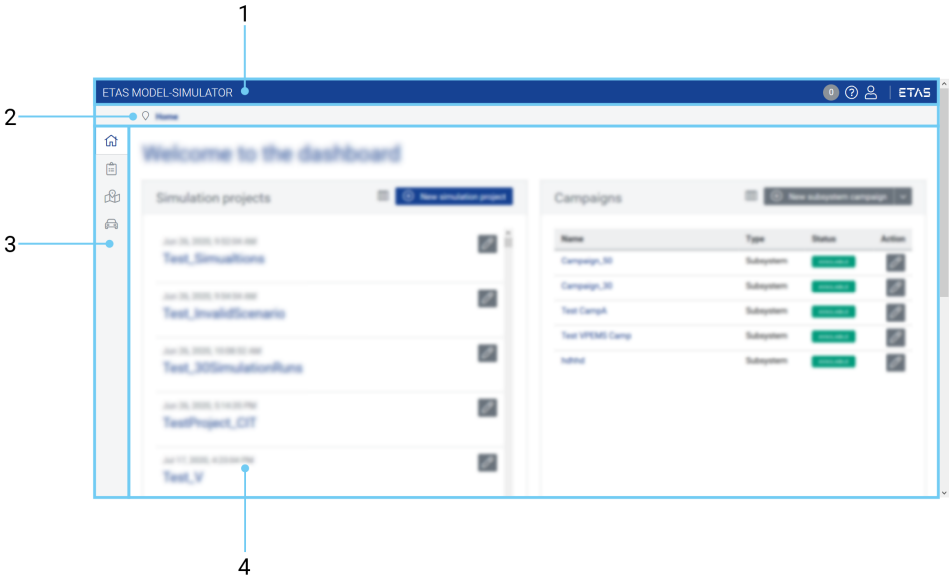


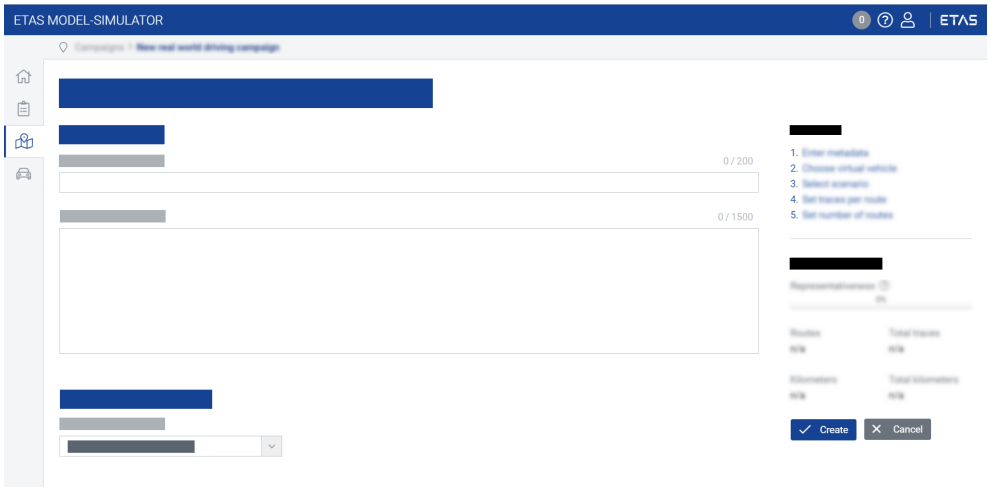
Fig. 5-1: Overview of user interface areas

After signing in, you get to the home screen with your dashboard showing an extract of the latest simulation projects, test execution projects, campaigns and models each in a separate tile.

Every menu item of the navigation bar contains the possibility to open an overview or to create an item according to the selection.

5.4 Input Forms

For creating projects, campaigns or models input forms are used. This supports you to easily create items on one page. The input forms are designed similarly. The campaigns section gives you the possibility to access the entry sections via the summarized steps on the right instead of just scrolling down.



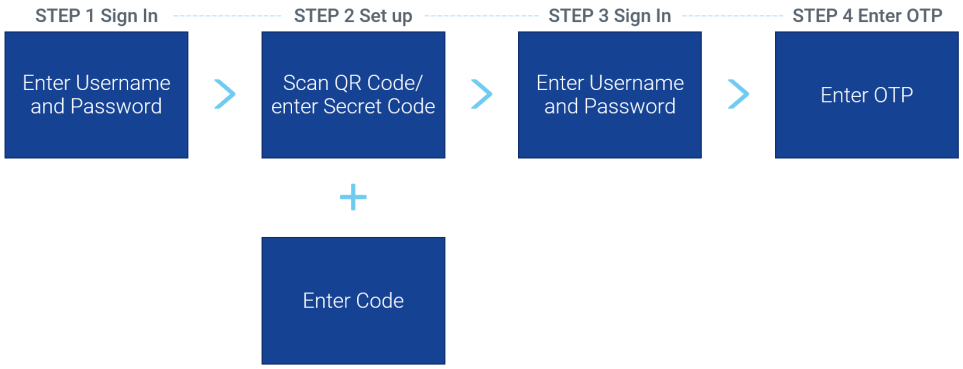
5.5 Multi-Factor Authentication

The multi-factor authentication provides a secure way to sign in to the MODEL-SIMULATOR. Every sign in you have to authenticate your user with an authentication code.

That means you need a generated password in addition to the self-created password. For the authentication you need an application on your mobile phone, mobile device or on your PC that generates a time-based one-time password (TOTP). There are several authentication apps on the market which are working very similar. They can be found in the internet or app stores using the keywords "Authentication" or "TOTP" for example.

5.5.1 Set up TOTP Authentication

For setting up the **t**ime-based **o**ne-**t**ime **p**assword (TOTP) authentication the following workflow is designed.



✓ Prerequisite: Authentication app is downloaded on mobile device or PC and ready to use.

1. Enter username and password in the MODEL-SIMULATOR.
2. Scan QR code with the TOTP application on your device.

or

Enter secret code in the TOTP application on your device.

⇨ A TOTP code will be shown.

3. Enter this code in the **OTP code** field.

⇨ Sign In page is shown.

4. Enter username and password .
5. Enter this code in the **OTP code** field.

⇨ The MODEL-SIMULATOR dashboard is shown.

5.5.2 Sign in via TOTP

1. Open TOTP application on PC or mobile device.
2. Go to Sign in page of the MODEL-SIMULATOR.
3. Enter username and password.
4. Enter the code you received via TOTP application.

⇨ The MODEL-SIMULATOR dashboard is shown.

5.6 Header

In the header you can check current processes **(1)** or open support section **(2)**. With the user profile symbol **(3)** you can change your password and sign out. The ETAS logo brings you to the ETAS website **(4)**, the breadcrumb **(5)** shows where you are and allows you to use for navigation.

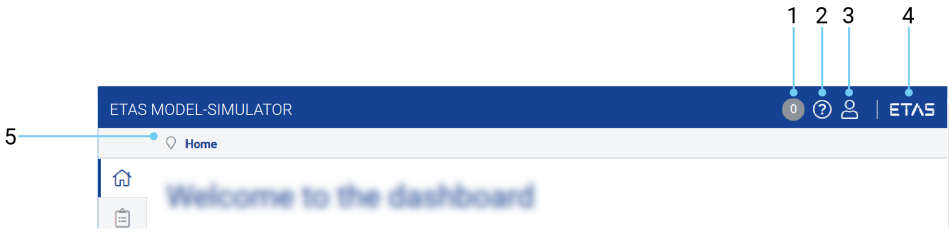


Fig. 5-2: Overview of header areas

The header section provides you several features:

ETAS MODEL-SIMULATOR	Opens home section with your dashboard.
0	Shows current background actions and their progress: 3 blue symbolizes an ongoing action, 1 red symbolizes a failed action, 4 yellow shows a partly successful action, and 2 green shows a successful procedure. By clicking on it you get more details. The number indicates number of current actions.
	Opens support section with access to User documentation, Support contact, Data protection notice, Imprint or Safety Advice and more.
	Opens my account section where you can change your password or sign out.
ETAS	Opens ETAS website in a new tab.

5.6.1 Change Password

Quick Access:

(header) › Change password

1. Go to My account → **Change password** in the header.
   ⇨ Change password site is displayed.
2. Enter required fields and **Change password**.

5.7 Home

The home section with your dashboard gives you an overview of the latest:

- simulation projects **(1)**
- test execution projects **(2)**
- campaigns **(3)**
- models **(4)**

see Fig. 5-3: "Overview of dashboard (home) areas" on the next page

It shows a selection of each, within every tile you can directly jump into the sections and elements.

The dashboard combines many functionalities of the MODEL-SIMULATOR, that means you can use it as central control and it is the fastest way to execute most features except simulations. Within every section you can get to the overviews or the details of a selected item and create a simulation project, test execution project, campaign or model. Simulations are located in a separate tab in a simulation project. Test executions are located in a separate tab in a test execution project.

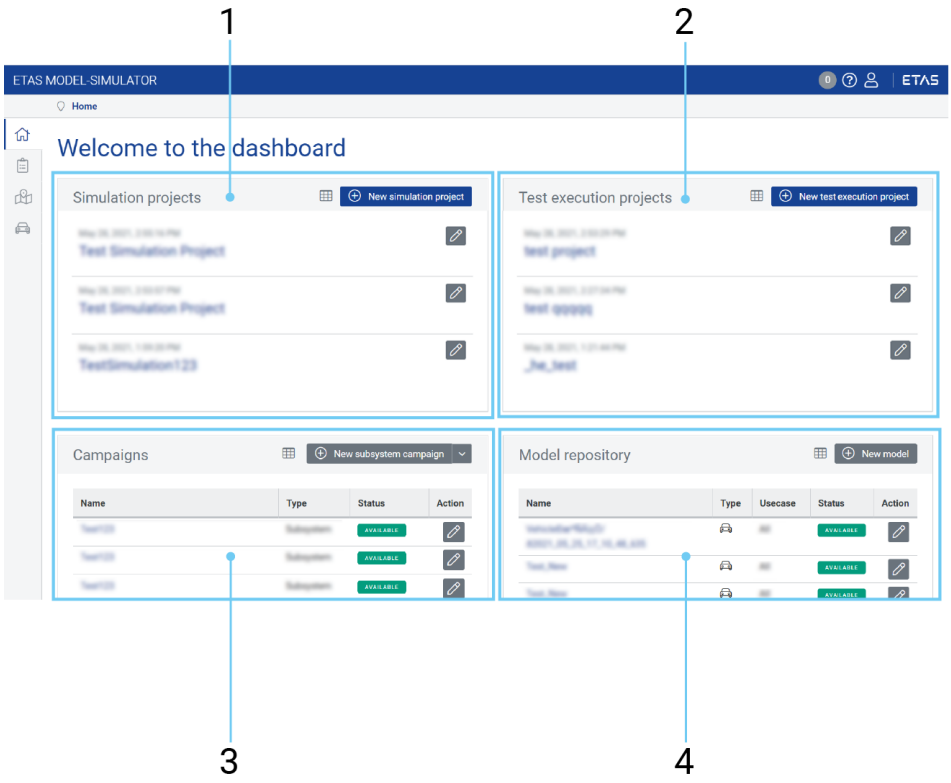
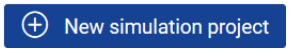


Fig. 5-3: Overview of dashboard (home) areas

The home section provides you several features:

	Opens overview of selected item (simulation projects, campaigns or models).
	Opens detail site of selected item (simulation project, campaign or model). You can also click on a simulation project, campaign or modelname to get to the detail site. Afterwards you have the possibility to edit details.
	Opens simulation projects section with entry form where you can create a new simulation project.

New test execution project	Opens test execution projects section with entry form where you can create a new test execution project.
New subsystem campaign	Opens campaigns section with entry form where you can create a new subsystem campaign. Use split button to select virtual vehicle campaign, test execution campaign for campaign creation.
New model	Opens models section with entry form where you can create a new model.

5.8 Simulation Projects

A Simulation project contains the following items.

- Reports: show report information and download reports
- Simulations (including campaigns): show and manage simulations (add, run, stop, download or delete simulations), see further information like status, progress, predicted costs and predicted time
- Models: show used models (type and status)
- Project details: show and edit simulation project information

Every item can be addressed via a separate tab in an opened simulation project.

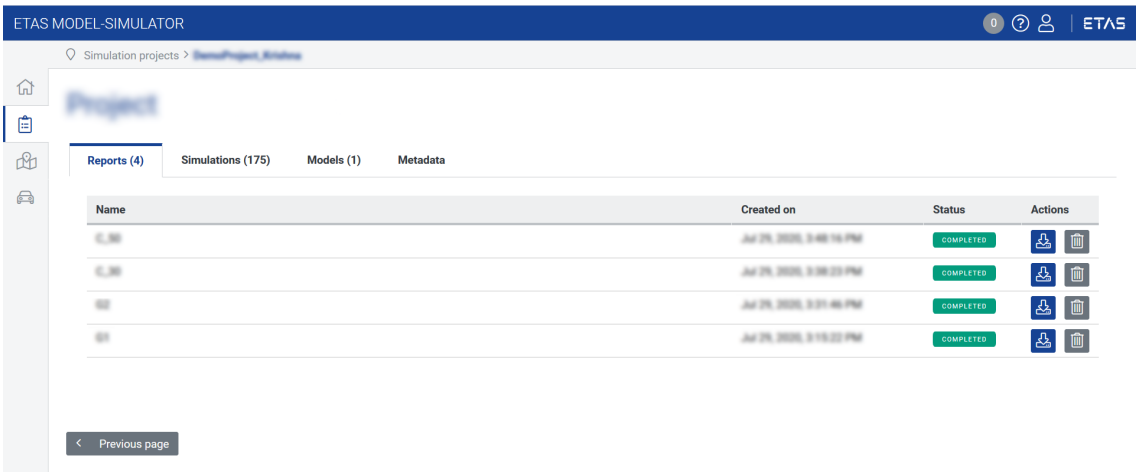


Fig. 5-4: Simulation project details (screenshot)

If you are interested in conceptual background information, check 3.3.1 "Simulation Projects" on page 14.

5.8.1 Open Simulation Projects Overview

Quick Access:

Projects *(navigation bar)* › *Simulation projects overview*

1. Go to **Projects** → **Overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Open simulation project by clicking on name or using to get to the detail site.

5.8.2 Search Simulation Projects

Quick Access:

Projects (navigation bar) › Simulation projects overview › Filter projects

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All projects are displayed.
2. Type in your keyword into the search box (`Fiter simulation projects`) to filter list of all projects.
   ⇨ All columns are searched while typing and results are displayed.

Search operators like "", *, AND, etc. **cannot** be used.

5.8.3 Filter Simulation Projects Overview

Quick Access:

Projects (navigation bar) › Simulation projects overview

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Click on dropdown (`Limit filter columns`) to limit columns to which the search filter should be applied.
   ⇨ Selected columns are searched while typing and results are displayed.

Search operators like "", *, AND, etc. cannot be used.

5.8.4 Sort Simulation Projects Overview

Quick Access:

Projects (navigation bar) › Simulation projects overview ›

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Click on column heading to sort simulation projects ascending and descending alphabetically or by time (column `created on`).
   ⇨ Selected columns are sorted. When sorting, only the selected column is considered, the other columns are neglected.

5.8.5 Create a Simulation Project

Quick Access:

Projects *(navigation bar)* › *New simulation project*

1. Go to **Projects** → **New simulation project** in the navigation bar.
2. Enter required fields:
   › Project name
   › Description (optional)
   › Select a model out of the list
   › Select a report template out of the list
3. Click **Create**.
   ⇨ Simulation project is created and displayed on top of **Simulation projects overview**.

5.8.6 Delete a Simulation Project

Quick Access:

Projects *(navigation bar)* › *Simulation projects overview* ›

1. Go to **Projects** → **Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Click of the simulation project you want to delete (`Actions` column) and confirm dialog.
   ⇨ Simulation project containing all simulations, simulation results and reports will be deleted.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

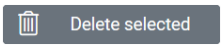
5.8.7 Delete Multiple Simulation Projects

Quick Access:

Projects *(navigation bar)* › *Simulation projects overview* › ›

Delete selected

1. Go to **Projects** → **Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Select multiple simulation projects by activating the check boxes of simulation projects you want to delete.

3. Click 

and confirm dialog.
   ⇨ Simulation projects containing all simulations, simulation results and reports will be deleted.

> **Note**
>
> You can select all simulation projects by using the checkbox in the table header. If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.8.8 Edit a Simulation Project

Quick Access:

Projects *(navigation bar)* › *Simulation projects overview* ›

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Select simulation project you want to edit by clicking on simulation project name or using .
   ⇨ Project details are opened.
3. Go to **Project details** tab.
4. To adjust simulation project name or Description select **Edit**.
5. Click **Save**.

5.8.9 Open Simulation Project Details

Quick Access:

Projects *(navigation bar)* › *Simulation projects overview* ›

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Select simulation project you want to open by clicking on simulation project name or using .
   ⇨ Project details are opened.
3. Browse through the tabs.

5.8.10 Add Simulation Group(s) to a Simulation Project

Quick Access:

Projects *(navigation bar)* › *Simulation projects overview* › › *Simulations tab*

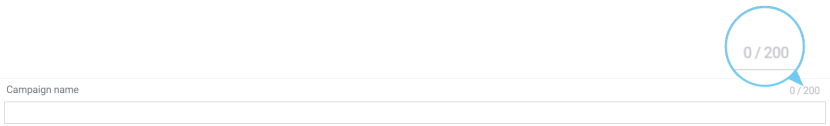
✓ Prerequisite: simulation project created. (See Create a Simulation Project)
✓ Prerequisite: campaign(s) created. (See Create a Virtual Vehicle Campaign or Create a Subsystem Campaign)

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Select simulation project you want to provide with simulation(s) by clicking on simulation project name or using .
   ⇨ Project details are opened.
3. Go to **Simulations** tab.
4. Click **New simulation group** .
   ⇨ New simulation group entry form is opened.
5. Enter required fields:
   › Simulation group name
   › Description (optional)
   › Select campaign(s) with double click

> **Note**
>
> You can select more than one campaign by using STRG/CTRL key and . If you want to select **all** campaigns, use .

6. Click **Create**.
   ⇨ Simulation group is created and displayed on top of **Simulations** tab.
   Due to character limitation of 140 characters for simulation group name, the name will be truncated. Each component can be up to 44 characters long. [Campaign_name]_[Folder_name]_[File_name]_[Sequence_number]
   To ensure that the file name is not truncated, make sure that campaign name, files names of *.zip and *.mf4 are not longer than 44 characters.

   Used characters are shown during creation at the right edge of the text input field.



5.8.11 Run Simulations in a Simulation Project

Quick Access:
Projects *(navigation bar)* › *Simulation projects overview* › › *Simulations tab* ›

✓ Prerequisite: simulation project created. (See Create a Simulation Project)
✓ Prerequisite: simulation group(s) created and labeled with READY. (See Add Simulation Group(s) to a Simulation Project)

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Select simulation project by clicking on simulation project name or using .
   ⇨ Simulation project details are opened.
3. Go to **Simulations** tab.
4. Select to start simulation.
   ⇨ Simulation is started. Once a simulation group is completed, a report will be generated automatically.

> **Note**
>
> You can also start single simulations of the simulation group. Expand simulation group and click individual start button of the simulation level (simulation run or simulation job).

For information on status labels see 3.11.2 "Simulation Status and Test Execution Status" on page 42 and 3.11.5 "Tooltips for Campaign Status" on page 45.

5.8.12 Stop Simulations in a Simulation Project

Quick Access:
Projects *(navigation bar) › Overview* *›* *› Simulations tab ›*

> **Note**
>
> Simulations can just be stopped within a certain time frame.

✓ Prerequisite: simulation project created. (See Create a Simulation Project)
✓ Prerequisite: simulation group(s) created and labeled with RUNNING. (See Add Simulation Group(s) to a Simulation Project)

1. Go to **Projects → Overview** in the navigation bar.
2. Select simulation project by clicking on simulation project name or using .
   ⇨ Project details are opened.
3. Go to **Simulations** tab.
4. Select to stop selected simulation on e.g. simulation run level.
   ⇨ Simulation is stopped and labeled with STOPPED
   simulation job and simulation group are labeled with INCOMPLETE.

See also 3.11.2 "Simulation Status and Test Execution Status" on page 42.

5.8.13 Delete a Simulation Group in a Simulation Project

Quick Access:

Projects (navigation bar) › Simulation projects overview › › Simulations tab ›

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Select simulation project by clicking on simulation project name or using .
   ⇨ Project details are opened.
3. Go to **Simulations** tab.
4. Click of the simulation group and confirm dialog.
   ⇨ Simulation group and containing simulation job(s) and simulation run (s) as well as containing simulations are deleted. Reports created on the deleted simulation group remain unaffected and can be still downloaded.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.8.14 Delete a Simulation Job in a Simulation Project

Quick Access:

Projects (navigation bar) › Simulation projects overview › › Simulations tab ›

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Select simulation project by clicking on simulation project name or using .
   ⇨ Project details are opened.
3. Go to **Simulations** tab.
4. Expand simulation group to see containing simulation jobs.
5. Click of the simulation job and confirm dialog.
   ⇨ Simulation job and containing simulation runs are deleted.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.8.15 Delete a Simulation Run in a Simulation Project

Quick Access:

Projects *(navigation bar)* › *Simulation projects overview* › › *Simulations tab* ›

1. Go to **Projects** → **Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Select simulation project by clicking on simulation project name or using .
   ⇨ Project details are opened.
3. Go to **Simulations** tab.
4. Expand simulation group and simulation job to see containing simulation runs.
5. Click of the simulation run and confirm dialog.
   ⇨ Simulation run is deleted.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.8.16 Create a Report

Quick Access:

Projects › *overview* *(navigation bar)* › › *Reports tab* › *New report*

✓ Prerequisite: project created. (See Create a Simulation Project"Create a Test Execution Project" on page 78)
✓ Prerequisite: simulation group labeled with COMPLETED.
✓ Prerequisite: report template created and labeled with AVAILABLE. (See Create a Report Template)

1. Go to **Projects** → **Overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Open simulation project by clicking on name or using .
3. Click **New report** in **Reports** tab.
4. Enter required fields:
   › Report name
   › Description (optional)
5. Select **simulation group** out of the list.
6. Select **report template** out of the list.

7. Click **Create**.
   ⇨ Created report is shown in the Reports tab and labeled with PROCESSING. Once the report is ready, it is labeled with COMPLETED and can be downloaded.

5.8.17 Download a Report

Quick Access:
Projects (navigation bar) › Simulation projects overview › › Reports tab ›

✓ Prerequisite: Simulation finished and simulation group labeled with COMPLETED.
✓ Prerequisite: report is available and labeled with COMPLETED.

1. Go to **Projects → Simulation projects overview** in the navigation bar.
2. Select simulation project you want to open by clicking on simulation project name or using .
   ⇨ Project details are opened, **Reports** tab is shown. Reports are generated automatically on simulation group level, when the simulation is completed.
3. Click to download the report.
   ⇨ Simulation run name is used as the simulation run result name. If the run name has any white spaces or special characters those will be replaced by equivalent characters and be reflected in the result files.

5.8.18 Download Simulation Results

Quick Access:
Projects (navigation bar) › Simulation projects overview › › Simulations tab ›

✓ Prerequisite: Simulation finished and simulation group labeled with COMPLETED.

1. Go to **Projects → Simulation projects overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Select simulation project you want to open by clicking on simulation project name or using .
   ⇨ Project details are opened, **Reports** tab is shown.
3. Go to **Simulations** tab.

4. Click to download the simulation resultson different levels.
   ⇨ Simulation run name is used as the simulation run result name. If the run name has any white spaces or special characters those will be replaced by equivalent characters and be reflected in the result files.

5.8.19 Open a Report

Quick Access:

Projects *(navigation bar) › Overview* *›* *› Reports tab*

✓ Prerequisite: report is downloaded. (see 5.8.17 "Download a Report" on the previous page)

✓ Prerequisite: report is unzipped and saved.

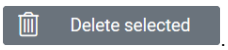
1. Open HTML file.
2. Drag & Drop folder as described in the notification.
   ⇨ Report is shown. You can browse through the views and visualized data.

5.8.20 Delete a Report

Quick Access:

Projects *› Overview* *›*

1. Go to **Projects** → **Overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Open simulation project by clicking on name or using .
3. Click of the report you want to delete (`Actions` column) and confirm dialog.

   You can also select and delete multiple reports using the checkbox and  .

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.9 Test Execution Projects

A Test execution project contains the following items.

- Tests (including campaigns): show and manage tests (add, run, stop or delete tests), see further information like execution status and result status
- Models: show used models (type and status)
- Project details: show and edit test execution project information

Every item can be addressed via a separate tab in an opened test execution project.

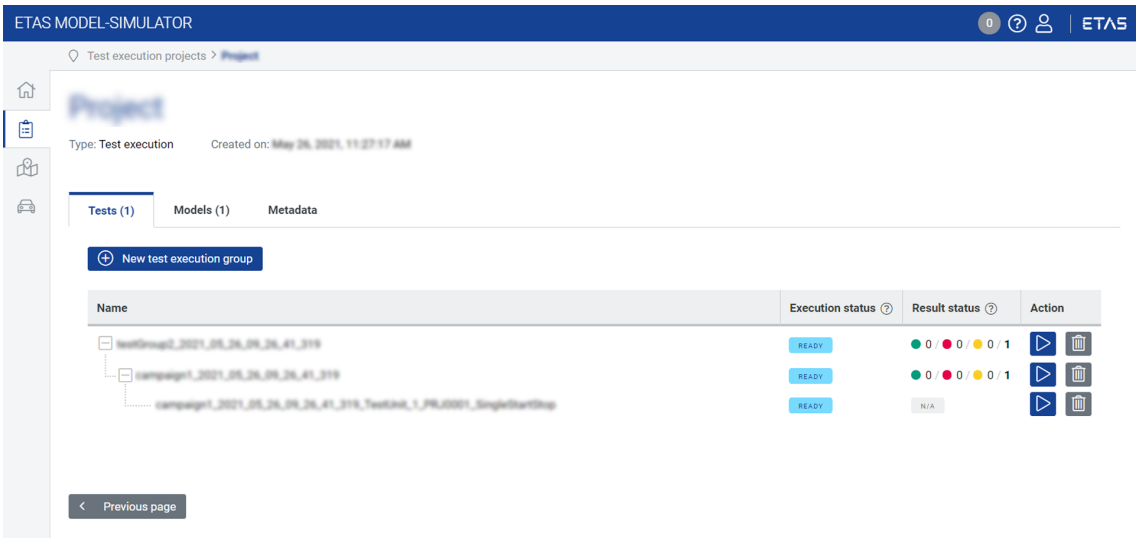


Fig. 5-5: Test execution project details (screenshot)

If you are interested in conceptual background information, check 3.3.3 "Test Execution Projects" on page 18.

5.9.1 Open Test Execution Projects Overview

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview*

1. Go to **Test execution projects → Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Open test execution project by clicking on name or using to get to the detail site.

5.9.2 Search Test Execution Projects

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview* › Filter projects

1. Go to **Test execution projects → Test execution projects overview** in the navigation bar.
   ⇨ All projects are displayed.

2. Type in your keyword into the search box (`Filter projects`) to filter list of all projects.
   ⇨ All columns are searched while typing and results are displayed.
   Search operators like "", *, AND, etc. **cannot** be used.

5.9.3 Filter Test Execution Projects Overview

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview*

1. Go to **Test execution projects → Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Click on dropdown (`Limit filter columns`) to limit columns to which the search filter should be applied.
   ⇨ Selected columns are searched while typing and results are displayed.
   Search operators like "", *, AND, etc. cannot be used.

5.9.4 Sort Test Execution Projects Overview

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview* ›

1. Go to **Test execution projects → Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Click on column heading to sort simulation projects ascending and descending alphabetically or by time (column `created on`).
   ⇨ Selected columns are sorted. When sorting, only the selected column is considered, the other columns are neglected.

5.9.5 Create a Test Execution Project

Quick Access:

Test execution projects *(navigation bar)* › *New test execution project*

1. Go to **Test execution projects → New test execution project** in the navigation bar.
   ⇨ All test execution projects are displayed.

2. Enter required fields:
   › Test execution project name
   › Description (optional)
   › Select one model with double click
3. Click **Create**.
   ⇨ Test execution project is created and displayed on top of **Test execution projects overview**.

5.9.6 Delete a Test Execution Project

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview* ›

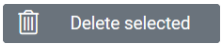
1. Go to **Test execution projects** → **Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Click of the test execution project you want to delete (`Actions` column) and confirm dialog.
   ⇨ Test execution project containing all tests and test results will be deleted.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.9.7 Delete Multiple Test Execution Projects

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview* ›

1. Go to **Test execution projects** → **Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Select multiple test execution projects by activating the check boxes of test execution projects you want to delete.
3. Click 

and confirm dialog.
   ⇨ Test execution project containing all tests and test results will be deleted.

> **Note**
>
> You can select all simulation projects by using the checkbox in the table header. If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.9.8 Edit a Test Execution Project

Quick Access:

Test execution projects *(navigation bar) › Test execution projects overview* ›

1. Go to **Test execution projects → Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Select test execution project you want to edit by clicking on test execution project name or using .
   ⇨ Project details are opened.
3. Go to **Project details** tab.
4. To adjust test execution project name or Description select **Edit**.
5. Click **Save**.

5.9.9 Open Test Execution Project Details

Quick Access:

Test execution projects *(navigation bar) › Test execution projects overview* ›

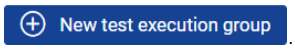
1. Go to **Test execution projects → Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Select test execution project you want to open by clicking on test execution project name or using .
   ⇨ Project details are opened.
3. Browse through the tabs.

5.9.10 Add Test Execution Group(s) to a Test Execution Project

Quick Access:

Test execution projects *(navigation bar) › Test execution projects overview* › › *Tests tab*

✓ Prerequisite: test execution project created. (See Create a Test Execution Project)
✓ Prerequisite: campaign(s) created. (See "Create a Test Execution Campaign" on page 92)

1. Go to **Test execution projects → Test execution projects overview** in the navigation bar.
2. Select test execution project you want to provide with test(s) by clicking on test execution project name or using .
   ⇨ Project details are opened.
3. Go to **Tests** tab.
4. Click **New test execution group** 

   ⇨ New test execution group entry form is opened.
5. Enter required fields:
   › Test execution group name
   › Description (optional)
   › Select campaign(s) with double click

> **Note**
>
> You can select more than one campaign by using STRG/CTRL key and . If you want to select **all** campaigns, use .

6. Click **Create**.
   ⇨ Test execution group is created and displayed on top of **Tests** tab.

5.9.11 Run Test Executions in a Test Execution Project

Quick Access:

Test execution projects *(navigation bar)* *› Test execution projects overview* *›* *› Tests tab ›*

✓ Prerequisite: test execution project created. (See Create a Test Execution Project)
✓ Prerequisite: test execution group(s) created and labeled with READY. (See Add Test Execution Group(s) to a Test Execution Project)

1. Go to **Test execution projects → Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Select test execution project by clicking on test execution project name or using .

⇨ Test execution project details are opened. **Tests** tab is shown.

3. Select ▷ to start test execution.
   ⇨ Test execution is started.

> **Note**
>
> You can also start single tests of the test execution group. Expand test execution group ⊞ and click individual start button of the test level (test execution run or test execution job).

For information on status labels see 3.11.2 "Simulation Status and Test Execution Status" on page 42 and 3.11.5 "Tooltips for Campaign Status" on page 45.

5.9.12 Stop Test Executions in a Test Execution Project

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview* › ✎ › *Tests tab* › □

> **Note**
>
> Test executions can just be stopped within a certain time frame.

✓ Prerequisite: test execution project created. (See Create a Test Execution Project)
✓ Prerequisite: test execution group(s) created and labeled with RUNNING. (See Add Test Execution Group(s) to a Test Execution Project)

1. Go to **Test execution projects → Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Select test execution project by clicking on test execution project name or using ✎.
   ⇨ Project details are opened. **Tests** tab is shown.
3. Select □ to stop selected test execution on e.g. test execution run level.
   ⇨ Test execution is stopped and labeled with STOPPED, test execution job and test execution group are labeled with INCOMPLETE.

See also 3.11.2 "Simulation Status and Test Execution Status" on page 42.

5.9.13 Delete a Test Execution Group in a Test Execution Project

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview* › › *Tests tab* ›

1. Go to **Test execution projects** → **Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Select test execution project by clicking on test execution project name or using .
   ⇨ Project details are opened. **Tests** tab is shown.
3. Click of the test execution group and confirm dialog.
   ⇨ Test execution group and containing test execution job(s) and test execution run(s) as well as containing tests are deleted.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.9.14 Delete a Test Execution Job in a Test Execution Project

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview* › › *Tests tab* ›

1. Go to **Test execution projects** → **Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Select test execution project by clicking on test execution project name or using .
   ⇨ Project details are opened. **Tests** tab is shown.
3. Expand test execution group to see containing test execution job(s).
4. Click of the test execution job and confirm dialog.
   ⇨ Test execution job and containing test execution runs are deleted.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.9.15 Delete a Test Execution Run in a Test Execution Project

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview* › › *Tests tab* ›

1. Go to **Test execution projects** → **Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Select test execution project by clicking on test execution project name or using .
   ⇨ Project details are opened. **Tests** tab is shown.
3. Expand test execution group and test execution job(s) to see containing test execution run(s).
4. Click of the test execution run and confirm dialog.
   ⇨ Test execution run is deleted.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.9.16 Download Test Execution Results

Quick Access:

Test execution projects *(navigation bar)* › *Test execution projects overview* › › *Tests tab* ›

✓ Prerequisite: Test execution finished and test execution group labeled with COMPLETED.

1. Go to **Test execution projects** → **Test execution projects overview** in the navigation bar.
   ⇨ All test execution projects are displayed.
2. Select test execution project you want to open by clicking on test execution project name or using .
   ⇨ Project details are opened, **Tests** tab is shown.
3. Click to download the test execution result on different levels.

5.10 Campaigns

A Campaign contains the following items.

- Campaign details: show and edit campaign information
- Campaign properties: show details of testing tool (only for test execution campaign)
- Files: show details and download files.
  - parameters (*.cdfx) and measurement (*.mf4) files of a subsystem campaign
  - measurement (*.mf4) files of a virtual vehicle campaign
  - testunit (*.prj) files of a test execution campaign

  Reason for unavailability is shown.

Every item can be addressed via a separate tab in an opened campaign.

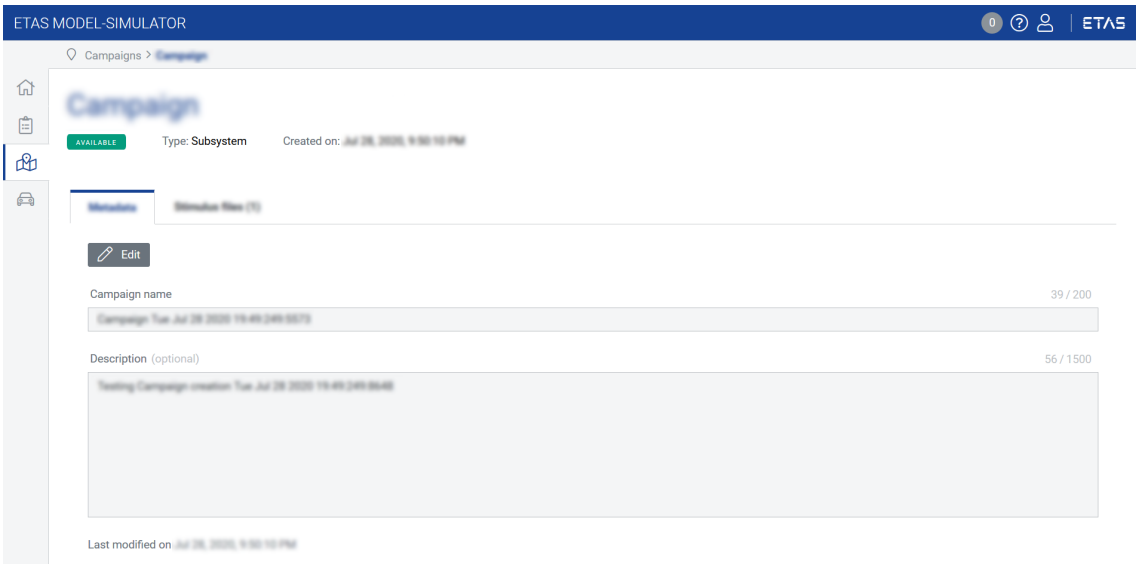
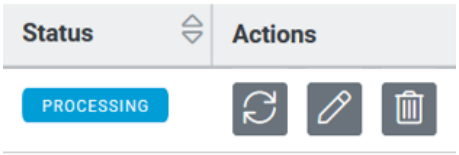
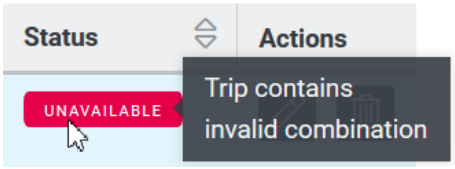


Fig. 5-6: Campaign details (screenshot)

If you are interested in conceptual background information, check 3.4 "Campaigns" on page 21.

5.10.1 Status Labels for Campaigns

The overview and opened campaigns show also the status:

Label	Description
AVAILABLE	Campaign is created successfully and ready to use in simulations.
PROCESSING	Campaign is created, but some process in the background is running (e.g. unzipping of files). Refresh and check the latest status, using the refresh button in Campaigns overview. 
UNAVAILABLE	Campaign is created, but some background processing failed (e.g. unzipping). Campaign is not available and cannot be used for simulations. See more information in tooltips (hover over labels) in Campaigns overview. Check the reason for unavailability and create a new campaign. 

Tab. 5-1: Overview status labels for campaigns

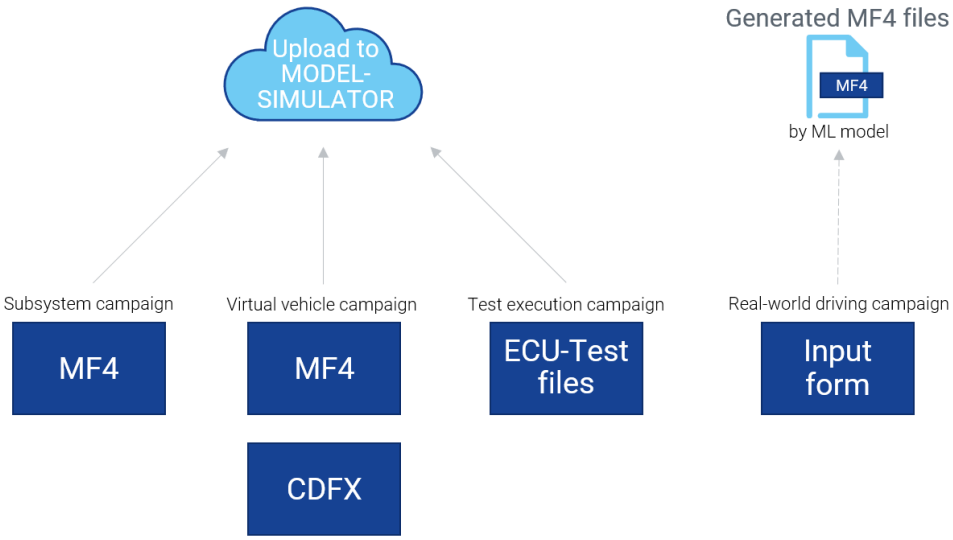
See also 3.11.5 "Tooltips for Campaign Status" on page 45.

5.10.2 Required Preprocessing Steps for Campaigns

There are 4 campaign types available in the MODEL-SIMULATOR:

- subsystem campaigns
- virtual vehicle campaigns
- test execution campaigns

For subsystem campaigns you just have to be sure to upload MF4 files in Version 4.1 (ASAM Standard for MDF) and to meet 3.10 "Stimuli File Requirements (*.mf4)" on page 40.



5.10.2.1 Preprocessing Steps for Virtual Vehicle Campaigns

1. Select the stimuli files on the PC (MDF files as *.mf4, Version 4.1, ASAM Standard for MDF).
2. Optional: Create the parameter file on the PC (*.cdfx, Version 3.1 or 4.0, ASAM standard for CDF).
3. Configure the virtual vehicle campaign ZIP on the PC and check all files (see 3.9.4 "Required Data Structure of Virtual vehicle campaigns" on page 29).

5.10.2.2 Preprocessing Steps for Test Execution Campaigns

1. Create test and configuration in ECU-Test V2023.2.3.
2. Configure the test execution campaign ZIP on the PC and check all files (see 3.9.5 "Required Data Structure of Test Execution Campaigns" on page 31).

5.10.2.3 ECU-Test Support and Limitations for Test Execution Campaigns

On the tool side, the MODEL-SIMULATOR supports the following features of ECU-Test:

- ECU-Test Version: V2023.2.3
- Workspace Configuration
  - Active projects (PRJ file): Use only active projects as inactive projects are treated as active projects. If your workspace contains inactive products delete them in ECU-Test or in the data you upload in the MODEL-SIMULATOR. If you upload them anyway, they will be shown and used for test execution.

    ECU-Test: **Project editor: Context menu > Activate**

- Referenced projects/packages (PRJ/PKG file): Use artifacts available in workspace as external references are not supported. If you have references linked, include them into the workspace folder you use for upload into the MODEL-SIMULATOR. All data which should be used, must be available in the uploaded data (self-contained).

  In general the ECU-TEST-workspace has not to contain any external links to any artifacts or to any folder (e.g. external folder for reports).

- Model access (TCF file): Use only one model file (e.g. MAPortConfig.xml) and configure one model access in TCF file as multiple running models in parallel are not yet supported. This used model has then be also selected in the test execution project.

  ECU-Test: **Test configuration editor: Platform tab > Model access node > Model file**

— Simulation Control

- Simulation Control configured in test bench configuration (TBC file): Start and Stop properties are supported
  > immediately on loading in TBC
  > with running the test case
  > no automatically simulation start or stop
  ECU-Test: **Test bench configuration editor: Select model node (Tools and ports) > Properties window**
- Multiple configuration changes in project (TBC/TCF file)

— Test Result (TRF file)

- Possible test result states (displayed in Result status column in Test execution projects)
  > SUCCESS: All test steps are passing with expected results.
  > FAILED: Test is failing due to not matching conditions.
  > ERROR: A tool error occured, or a test result is specified as ERROR.
  > INCONCLUSIVE: Trigger condition did not occur, no evaluation possible.
  ECU-Test: **Package editor: Signal recordings & Trace analysis tab**
  > NONE: No comparison of actual and expected value in the project available.
  ECU-Test: **Package editor: Test case tab**
  We show in the Result status the label you have set in the Test step Assertion and Test step result checkbox in ECU-Test, this test step can be found in the *Miscellaneous* folder of the *Default test steps* folder in the Actions window.

— Test Report (TCF file)

- Report configuration is supported (e.g. user-defined information or filters).
- Report format: All report formats are supported with the limitation that all formats without `TRFSPLITReportFormat` or only `TRFSPLITReportFormat` is working due to a ECU-Test defect (this works locally but not on cloud).
- Collection of LOG files (Content of Messages window of ECU-Test) are part of the generated report.

– Timing (TBC file)

  In the system two clocks are used: "wall clock" time (ECU-Test) and simulation time (Model). Depending on the model, the model runs faster than in real time. That means when you use "wait" in ECU-Test, it could be the model runs longer in simulation, because there is no synchronization. It must be considered, that these two clocks are not synchronized.

  - Using multiple measurements to check if a certain condition is fulfilled as time synchronizationis not supported.
  - Timeout of test execution (e.g. not meeting trigger condition): Take care about timeout-configuration in ECU-Test, there is no global timeout setting in MODEL-SIMULATOR available.
  - Not supported: Time synchronization between "wall clock"-time of ECU-Test and simulation time of simulation-ms datalogging via trigger condition (via mapping file).

– Interaction of Model created in COSYM and ECU-Test

  - ASAM XIL API supported (see ASAM XIL API Documentation in COSYM)

    ECU-Test: **Actions window: Jobs tab > Generic-XIL-API** and TBC file
  - Supported elements from COSYM model in ECU-Test
    - Standard COSYM modules (e.g. COSYM-CCode, Simulink): All elements like Inports, Outports, Measurements and Calibration variables are supported.
    - Imported modules (e.g. FMU): Only Inports and Outports are supported.

    ECU-Test: **Actions window: Model access tab**

See also Known Issue List in Release Notes for more specific information.

After the preprocessing you can create test execution campaigns and upload the data to the MODEL-SIMULATOR.

5.10.3 MF4 File Creation Tools

MDF4 Lib and MATLAB functions can be used to generate MF4 files automatically for Signals from Simulink. There are also tools available which record data in MDF format.

5.10.4 Modify Existing MF4 File

✓ Prerequisite: MDA readable tool available (e.g. ETAS MDA or ETAS ASCMO).

1. Open MF4 file.
2. Export MF4 file as ASCII.
3. Open ASCII file in Excel.
4. Change Data.
5. Import ASCII file.
6. Export selected signals as MF4.

You can also use the Calculated Signal function of MDA and extend the existing MF4 file.

5.10.5 Open Campaigns Overview

Quick Access:

Campaigns *(navigation bar)* › *Overview*

1. Go to **Campaigns** → **Overview** in the navigation bar.
   ⇨ All campaigns are displayed.
2. Open campaign by clicking on name or using to get to the detail site.

5.10.6 Search Campaigns

Quick Access:

Campaigns *(navigation bar)* › *Overview* › Filter campaigns

1. Go to **Campaigns** → **Overview** in the navigation bar.
   ⇨ All campaigns are displayed.
2. Type in your keyword into the search box (`Filter campaigns`) to filter list of all campaigns.
   ⇨ All columns are searched while typing and results are displayed. Search operators like "", *, AND, etc. **cannot** be used.

> **Note**
>
> You can also search for status labels with "available" for example. See also "Status Labels for Campaigns" on page 85 or "Campaign, Model and Report Templates Status" on page 44.

5.10.7 Filter Campaigns Overview

Quick Access:

Campaigns *(navigation bar)* › *Overview*

1. Go to **Campaigns** → **Overview** in the navigation bar.
   ⇨ All campaigns are displayed.
2. Click on dropdown (`Limit filter columns`) to limit columns to which the search filter should be applied.
   ⇨ Selected columns are searched while typing and results are displayed. Search operators like "", *, AND, etc. cannot be used.

5.10.8 Sort Campaigns Overview

Quick Access:

Campaigns (navigation bar) › Overview ›

1. Go to **Campaigns** → **Overview** in the navigation bar.
   ⇨ All campaigns are displayed.
2. Click on column heading to sort Campaigns ascending and descending alphabetically or by time (column `created on`).
   ⇨ Selected columns are sorted. When sorting, only the selected column is considered, the other columns are neglected.

5.10.9 Create a Subsystem Campaign

Quick Access:

Campaigns (navigation bar) › New subsystem campaign

✓ Prerequisite: Campaign data checked. (See Create a Simulation Project)

1. Go to **Campaigns** → **New subsystem campaign** in the navigation bar.
2. Enter required fields:
   › Campaign name
   › Description (optional)
3. **Upload**stimuli file(s) in *.mf4 format. (See "Check Campaign Data" on page 55 and 3.10 "Stimuli File Requirements (*.mf4)" on page 40)

   Note

   The status check and status display VALID or INVALID refers only to the file format not to its content.

4. Click **Create**.
   ⇨ Campaign is created and displayed in **Campaigns overview**. Upload progress of files can be seen in the header.

5.10.10 Create a Virtual Vehicle Campaign

Quick Access:

Campaigns *(navigation bar)* › *New virtual vehicle campaign*

✓ Prerequisite: Campaign data checked. (See Create a Simulation Project)

1. Go to **Campaigns** → **New virtual vehicle campaign** in the navigation bar.
2. Enter required fields:
   › Campaign name
   › Description (optional)
3. **Upload** campaign file in *.zip format with required structure. (See "Check Campaign Data" on page 55 and 3.10 "Stimuli File Requirements (*.mf4)" on page 40)

   > **Note**
   >
   > The status check and status display VALID or INVALID refers only to the file format not to its content.

4. Click **Create**.
   ⇨ Campaign is created and displayed in **Campaigns overview**. Upload progress of files can be seen in the header.

5.10.11 Create a Test Execution Campaign

Quick Access:

Campaigns *(navigation bar)* › *New test execution campaign*

✓ Prerequisite: Campaign data checked. (See Create a Simulation Project)

1. Go to **Campaigns** → **New test execution campaign** in the navigation bar.
2. Enter required fields:
   › Campaign name
   › Description (optional)
3. **Upload** ECU-Test data in *.zip format with required structure. (See "Check Campaign Data" on page 55)

   > **Note**
   >
   > The status check and status display VALID or INVALID refers only to the file format not to its content.

4. Click **Create**.
   ⇨ Campaign is created and displayed in **Campaigns overview**. Upload progress of files can be seen in the header.

5.10.12 Delete a Campaign

Quick Access:

Campaigns *(navigation bar)* › *Overview* ›

1. Go to **Campaigns** → **Overview** in the navigation bar.
   ⇨ All campaigns are displayed.
2. Click of the campaign you want to delete (`Actions` column) and confirm dialog.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.10.13 Delete Multiple Campaigns

Quick Access:

Campaigns *(navigation bar)* › *Overview* › › Delete selected

1. Go to **Campaigns** → **Overview** in the navigation bar.
   ⇨ All campaigns are displayed.
2. Select multiple campaigns by activating the check boxes of campaigns you want to delete.
3. Click Delete selected and confirm dialog.

> **Note**
>
> You can select all campaigns by using the checkbox in the table header. If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.10.14 Edit a Campaign

Quick Access:

Campaigns *(navigation bar)* › *Overview* ›

1. Go to **Campaigns** → **Overview** in the navigation bar.
2. Select campaign you want to edit by clicking on campaign name or using .
   ⇨ **Campaign details** are opened.
3. Select **Edit** to adjust campaign name or Description.

5.10.15 Open Campaign Details

Quick Access:

Campaigns *(navigation bar)* › *Overview* ›

1. Go to **Campaigns → Overview** in the navigation bar.
2. Select campaign you want to open by clicking on campaign name or using .
   ⇨ Campaign details are opened.
3. Browse through the tabs.

5.10.16 Use Campaign(s) in a Simulation Project

Quick Access:

Projects › *Overview* *(navigation bar)* › › *Simulations tab* › New simulation group

✓ Prerequisite: simulation project created. (See Create a Simulation Project)
✓ Prerequisite: campaign(s) created and labeled with AVAILABLE. (See Create a Virtual Vehicle Campaign or Create a Subsystem Campaign)

1. Go to **Projects → Overview** in the navigation bar.
2. Select simulation project by clicking on simulation project name or using .
   ⇨ Project details are opened.
3. Go to **Simulations** tab.
4. Click **New simulation group** New simulation group to add a simulation group.
   ⇨ Create new simulation group entry form is opened.
5. Enter required fields:
   › Simulation group name
   › Description (optional)
6. Select campaign(s) with double click.

> **Note**
>
> You can select more than one campaign by using STRG/CTRL key and ›. If you want to select **all** campaigns, use ».

7. Click **Create**.
   ⇨ Simulation group is created and shown in **Simulations** tab of simulation project.

5.10.17 Use Campaign(s) in a Test Execution Project

Quick Access:

Test execution projects › *Overview* *(navigation bar)* › › *Tests tab* › New test execution group

✓ Prerequisite: test execution project created. (See Create a Test Execution Project)
✓ Prerequisite: test execution campaign(s) created and labeled with AVAILABLE. (See Create a Test Execution Campaign)

1. Go to **Test execution projects → Overview** in the navigation bar.
2. Select test execution project by clicking on name or using .
   ⇨ Project details are opened.
3. Click **New test execution group** New test execution group to add a test execution group.
   ⇨ Create new test execution group entry form is opened.
4. Enter required fields:
   › Test execution group name
   › Description (optional)
5. Select campaign(s) with double click.

> **Note**
>
> You can select more than one campaign by using STRG/CTRL key and ›. If you want to select **all** campaigns, use ».

6. Click **Create**.
   ⇨ Test execution group is created and shown in **Tests** tab of test execution project.

5.10.18 Download Stimuli Files of a Campaign

Quick Access:

Campaigns *(navigation bar)* › *Overview* › › *Files tab* ›

1. Go to **Campaigns → Overview** in the navigation bar.
   ⇨ All campaigns are displayed.
2. Select campaign you want to open by clicking on campaign name or using .

⇨ Campaign details are opened.

3. Open **Files** tab.
4. Expand ⊞ data and click 📥 for download.

5.11 Models

A Model contains the following items.

- Model details: show and edit model information
- Model properties: show and edit real time factor
- Files: shows an overview of the uploaded data for the selected model. These files can be downloaded:
  - metadata (*.json)
  - deployables/executable (*.zip)
  - signal mapping (*.smf)
  - global parameter (*.cdfx)
  - datalogger configuration (*.dlc4)
  - signal interpolation (*.sti)

  Reason for unavailability is shown.

> **Note**
>
> Not all data may be available for download. That depends on what was uploaded while model creation.

Every item can be addressed via a separate tab in an opened model.

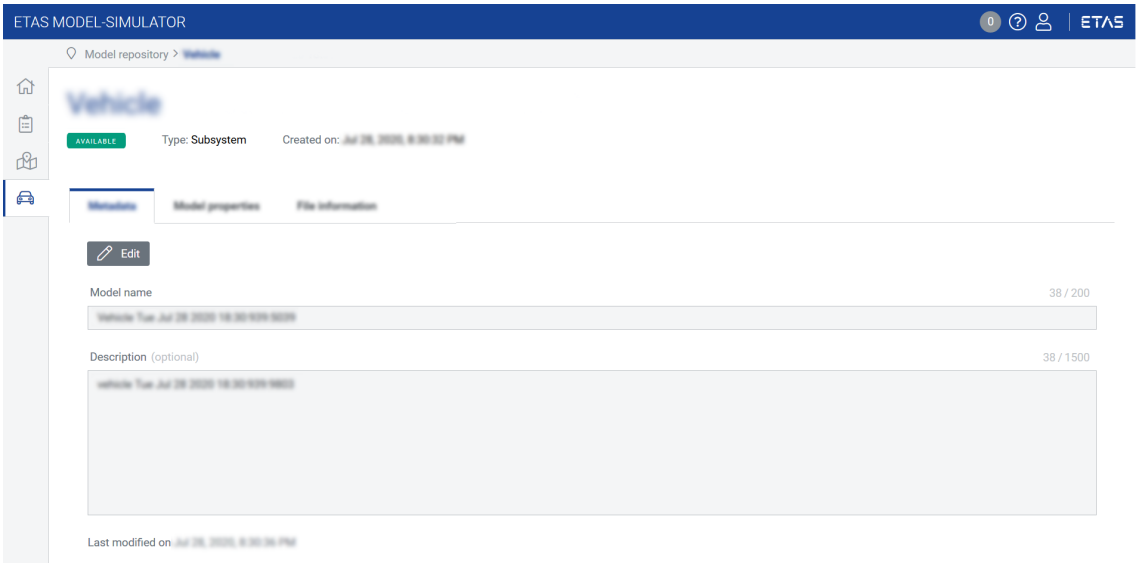
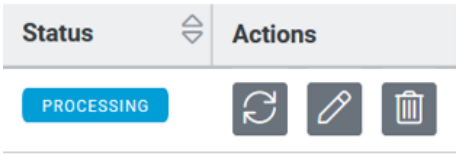
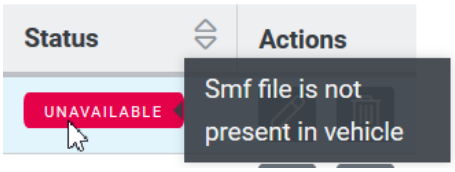


Fig. 5-7: Model details (screenshot)

If you are interested in conceptual background information, check 3.5 "Models" on page 21.

5.11.1 Status Labels for Models

The overview and opened models show also the status:

Label	Description
PROCESSING	Model is created, but some process in the background is running (e.g. unzipping of files). Refresh and check the latest status, using the refresh button in model repository. 
AVAILABLE	Model is created successfully and ready to use in simulation projects.
UNAVAILABLE	Model is created, but some background processing failed (e.g. unzipping). Model is not available and cannot be used for simulations. See more information in tooltips (hover over labels) in model repository. Check the reason for unavailability and create a new model. 

Tab. 5-2: Overview status labels for models

See also 3.11.6 "Tooltips for Model Status" on page 47.

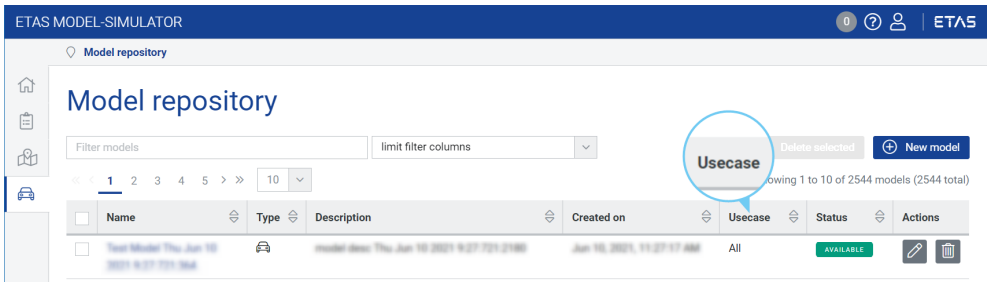
5.11.2 Use Cases

Depending on uploaded data, the following use cases are displayed in the model repository (usecase column):

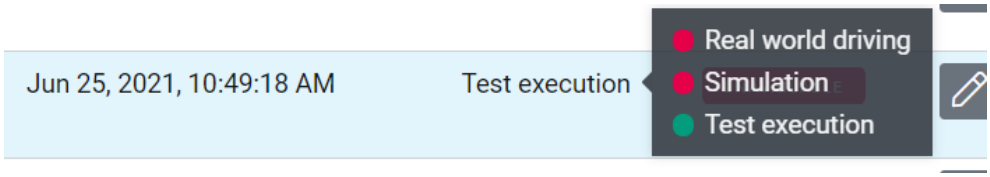
- All: *.zip with simConfig folder with SMF file and cloud_deployable.zip in COSYMprj folder is uploaded and can be used for simulation projects and test execution projects, see "Required Data Structure of Models (Use Case independent)" on page 35.
- Test: *.zip with cloud_deployable.zip in COSYMprj folder (without simConfig folder) is uploaded and can be used only for test execution projects, see "Required Data Structure of Models (Use Case independent)" on

page 35.

- None: not usable due to incorrect data upload or missing cloud_deployable.zip.



Hovering with the mouse over the **Usecase** type, the tooltip shows for which use case this model can be used.



In this example the model can only be used in a test execution projectcombined with a test execution campaign.

5.11.3 Required Preprocessing Steps for Models

There are some steps required for the data you have to upload into the Model repository.

1. Integrate models in COSYM on the PC (V3.2 for simulation projects, (V3.2 for test execution projects).
2. Build the deployables in COSYM on the PC. You can enable cloud deployable creation via **View › Settings** in COSYM. For more details see topics "Settings" and "Build" in COSYM User Guide or press F1 in COSYM and navigate to *User Interface › Settings* and *System Build › Build*.
3. Mandatory: Create the interpolation file (*.sti) with a text editor and the sample file format, see "Create a STI File" on page 100.
4. Optional: Create the signal mapping file (*.smf) with Sut Mapping File Editor (tool usually comes with COSYM).

   Note

   The Model's internal variable names have to match with the parameter file (*.cdfx) within the model ZIP file.

   <SHORT-NAME> tag of CDFX (within ‹SW-INSTANCE› block) file has to use the same name as the label in the mapping file (“SUTMap Testlabel" in the SMF file).

5. Optional: Create the parameterization file (*.cdfx) with COSYM, EE (Experiment Environment) or other ASAM CDF supported tool.

6. Optional: Create the datalogger configuration file (*.dlc4) with EE (Experiment Environment) → Datalogger tab.
7. Configure the model ZIP file on the PC and check all files (see 3.9.6 "Required Data Structure of Models" on page 34).

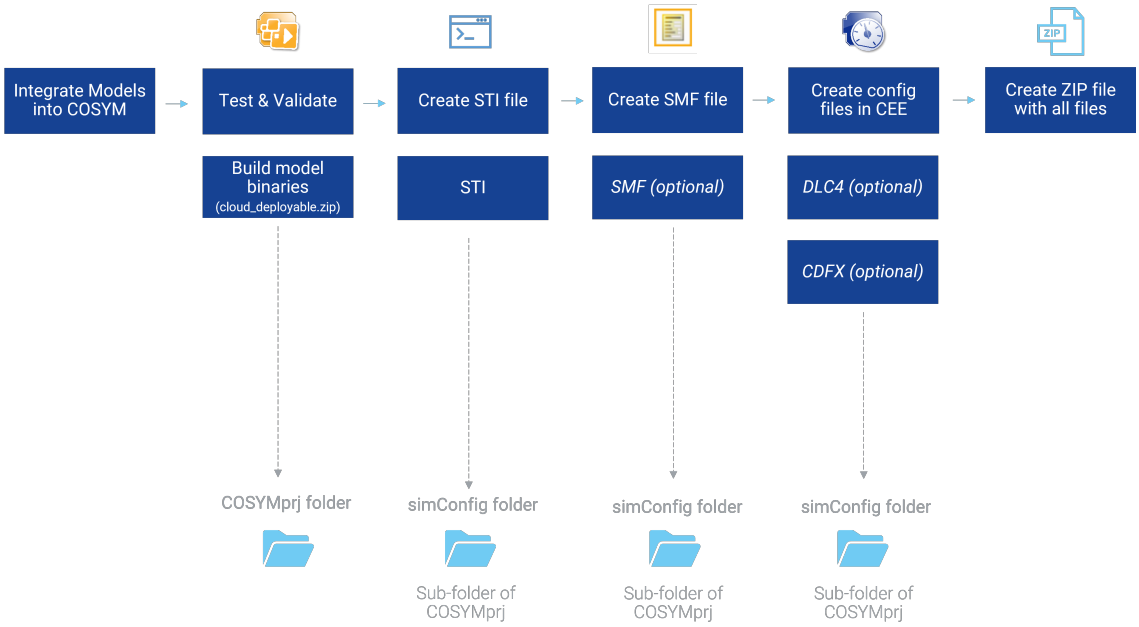


Fig. 5-8: Workflow required preprocessing steps for Models

After the preprocessing you can create and upload Models in the MODEL-SIMULATOR.

> **Note**
>
> Only the deployables (`cloud_deployable.zip`) can be generated with COSYM. The overall ZIP file and the remaining files and the folder inside must be created and added manually.

See 3.9.6 "Required Data Structure of Models" on page 34.

5.11.4 Models in COSYM

The following workflow steps are required to build a model in COSYM. For detailed information see seperate COSYM User Guide or press F1 within COSYM to open the online help.

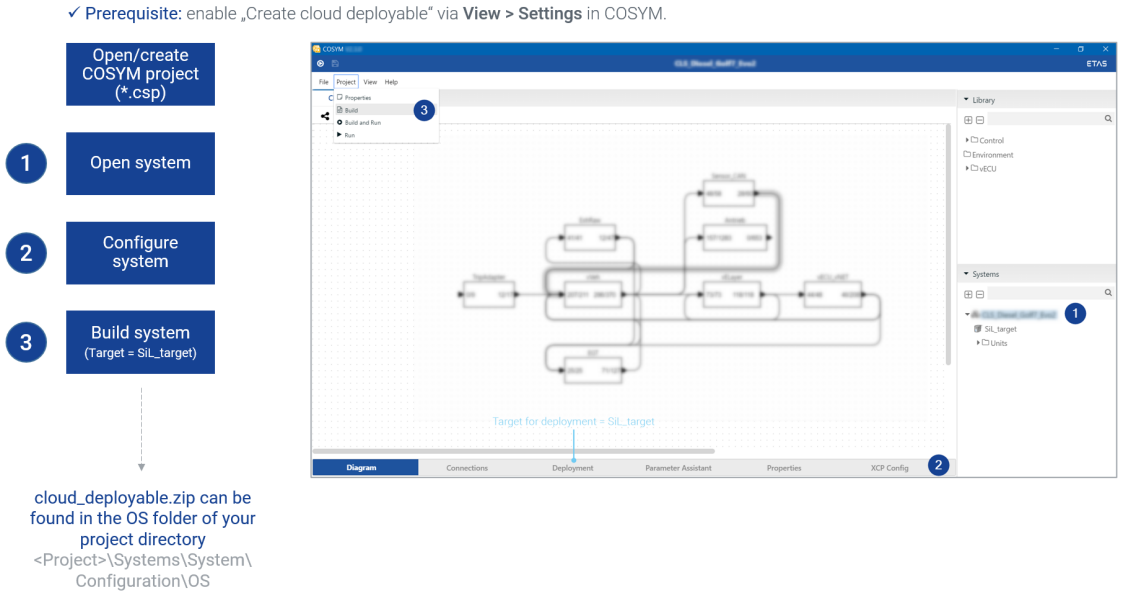


Fig. 5-9: Overview of steps in COSYM for model creation

5.11.5 Create a STI File

The STI file is a mandatory part of the model ZIP file (see Fig. 3-16 in 3.9.6 "Required Data Structure of Models" on page 34) and refers to the ASAM XIL standard.

1. Open a text editor.
2. Copy the given content below.
3. Enter your values.
4. If necessary copy the `SignalDescription` section for each signal of the stimuli MF4 file.
5. Go to **File > Save As** and save your file as STI. (*.sti).

Save that file in the simconfig folder (see Fig. 3-16 in 3.9.6 "Required Data Structure of Models" on page 34).

STI File Copy Content

```
<?xml version="1.0" encoding="utf-8"?>
<SignalDescriptionFile xmlns="http://www.asam.net/XIL/Signal/2.0" xmlns:xsi="h-
ttp://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="h-
ttp://www.asam.net/XIL/Signal/2.0 SignalDescriptionFormat.xsd" version="2.0">
<!--XIL API Signal Description File - generated by ETAS Experiment Environment,
ETAS GmbH-->
  <SignalDescriptionSet name="ECS">
  <!-- Beginning of paragraph <SignalDescription>. Copy for each signal. -->
  <!-- Enter the <SignalDescription name and id>, <DataVectorName> and <inter-
polation value>.-->
  <SignalDescription name="<!-- fill in here -->" id="<!-- fill in here -->">
   <SegmentSignalDescription>
    <DataFileSegment>
     <Duration xsi:type="ConstSymbol">
     <Value></Value> <!-- To be filled in by MODEL-SIMULATOR -->
     </Duration>
     <DataVectorName><!-- fill in here --></DataVectorName>
     <FileName></FileName> <!-- To be filled in by MODEL-SIMULATOR -->
     <Interpolation xsi:type="InterpolationType" value="<!-- fill in here-->"
```

```
/>
     <Start>
     <Value>0</Value>
     </Start>
     <TimeVectorName />
     <ChannelSource />
     <ChannelPath />
     <GroupName />
     <GroupSource />
     <GroupPath />
    </DataFileSegment>
   </SegmentSignalDescription>
  </SignalDescription>
  </SignalDescriptionSet>
   <!-- End of paragraph <signalDescription>. -->
   <!-- Link the <SignalDescription> to the signal in the stimuli MF4 file.-->
  <Assignments>
  <Link key="<!-- fill in here -->" value="<!-- fill in here-->" />
  </Assignments>
</SignalDescriptionFile>
```

STI is an XML-based data format and can be created with a text editor. See more information about STI on ASAM website: https://www.asam.net/standards/detail/xil/

5.11.6 Create a SMF File

1. Open Sut Mapping File/SMF Editor (e.g press the Windows key and start typing "Sut ..") .
2. Configure signal mapping file.
3. Go to **File > Save As** and save your file as SMF. (Smf File Version = Version 4.0, Smf File Version 3.1 = Version 3.1)

See also Sut Mapping File/SMF Editor User Guide (**Help > User Guide**) for more information.

SMF is an XML-based data format, see the basic structure:

```
<?xml version="1.0" encoding="utf-8"?>
<SutLabelMapping xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd= "http://www.w3.org/2001/XMLSchema" ext=".smf" ver="4.0" typ="SuT Map-
ping File">
    <ToolLabelList>
        <ToolLabelInfo />
            <DataTypeInfo />
    </ToolLabelList>
    <TestLabelList>
        <TestLabelMapper />
    </TestLabelList>
    <HierarchyList>
        <Hierarchy Path />
    </HierarchyList>
</SutLabelMapping>
```

See one example for Signal v_TargetSpeed (SMF Version 3.1)

```
<?xml version="1.0" encoding="utf-8"?>
<file xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ext=".smf" typ="SUT
Mapping File" ver="3.1" xsi:noNamespaceSchemaLocation= "http://www.etas-
group.com/downloads/xmlns/labcar_automation/lca_smf_v3.1.xsd">
```

```
    <!--Target Speed-->
    <SUTMap TestLabel="v_TargetSpeed" ToolLabel="TripAdapter/Inports/v_Stim_
kmph" Hierarchy="TripAdapter" SUTType="Measurement" ObjectType="Scalar"
DataType="Float" Comment=" Target Speed for Trip Adapter" Max="" Min="" Unit=""
XUnit="kmph" YUnit=""/>
</file>
```

5.11.7 Open Models Overview

Quick Access:

Models *(navigation bar)* › *Overview*

1. Go to **Models** → **Overview** in the navigation bar.
   ⇨ All Models are displayed.
2. Open model by clicking on name or using to get to the detail site.

5.11.8 Search Models

Quick Access:

Models *(navigation bar)* › *Overview* › Filter models

1. Go to **Models** → **Overview** in the navigation bar.
   ⇨ All Models are displayed.
2. Type in your keyword into the search box (`Filter models`) to filter list of all Models.
   ⇨ All columns are searched while typing and results are displayed. Search operators like "", *, AND, etc. cannot be used.

> **Note**
>
> You can also search for status labels with "available" or model type with "virtual" for virtual vehicle models for example. See also "Status Labels for Models" on page 97 or "Campaign, Model and Report Templates Status" on page 44.

5.11.9 Filter Models Overview

Quick Access:

Models *(navigation bar)* › *Overview*

1. Go to **Models** → **Overview** in the navigation bar.
   ⇨ All Models are displayed.

2. Click on dropdown (`Limit filter columns`) to limit columns to which the search filter should be applied.
   ⇨ Selected columns are searched while typing and results are displayed. Search operators like "", *, AND, etc. cannot be used.

5.11.10 Sort Models Overview

Quick Access:
Models *(navigation bar)* › *Overview* ›

1. Go to **Models** → **Overview** in the navigation bar.
   ⇨ All Models are displayed.
2. Click on column heading to sort models ascending and descending alphabetically or by time (column `created on`).
   ⇨ Selected columns are sorted. When sorting, only the selected column is considered, the other columns are neglected.

5.11.11 Create a Model

Quick Access:
Models *(navigation bar)* › *New model*

✓ Prerequisite: Model data checked. (See Create a Simulation Project)

1. Go to **Models** → **New model** in the navigation bar.
2. Select model type. (See "Models" on page 21 for more information.)
3. Enter required fields:
   › Model name
   › Real time factor
   › Description (optional)
4. Click **Upload** and select ZIP file. (See Check Model Data)

> **Note**
>
> The status check and status display VALID or INVALID refers only to the file format not to its content.

5. Click **Create**.
   ⇨ Model is created and displayed in **model repository** when upload was successful. Upload progress of files can be seen in the header.

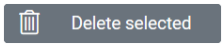
5.11.12 Delete a Model

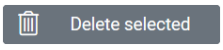
Quick Access:
Models *(navigation bar)* › *Overview* ›

1. Go to **Models → Overview** in the navigation bar.
   ⇨ All Models are displayed.
2. Click of the model you want to delete (`Actions` column) and confirm dialog.

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.11.13 Delete multiple Models

Quick Access:
Models *(navigation bar)* › *Overview* › › 

1. Go to **Models → Overview** in the navigation bar.
   ⇨ All models are displayed.
2. Select multiple models by activating the check boxes of models you want to delete.
3. Click  and confirm dialog.

> **Note**
>
> You can select all models by using the checkbox in the table header. If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.11.14 Edit a Model

Quick Access:
Models *(navigation bar)* › *Overview* ›

1. Go to **Models → Overview** in the navigation bar.
2. Select model you want to edit by clicking on model name or using .
   ⇨ **Model details** are opened.
3. Select **Edit** to adjust model name or description.
   Real time factor can be edited in the **Model properties** tab. See also "Edit Real Time Factor" below.

5.11.15 Edit Real Time Factor

Quick Access:
Models *(navigation bar)* › *Overview* ›

Based on run simulations a new RTF might be proposed which can be applied easily.

1. Go to **Models → Overview** in the navigation bar.
2. Select model you want to edit by clicking on model name or using .
   ⇨ Model details are opened, **metadata** tab is shown.
3. Go to **Model properties** tab.
4. Select **Edit** to adjust real time factor.

5.11.16 Open Model Details

Quick Access:
Models *(navigation bar)* › *Overview* ›

1. Go to **Models → Overview** in the navigation bar.
2. Select model you want to open by clicking on model name or using .
   ⇨ Model details are opened.
3. Browse through the tabs.

5.11.17 Use Model in a Project (Simulation Project or Test Execution Project)

Quick Access:
Projects *(navigation bar)* › *New simulation project / New test execution project*

✓ Prerequisite: project created. (See Create a Simulation Project or Create a Test Execution Project)
✓ Prerequisite: model created and labeled with AVAILABLE. (See Create a Model)

1. Go to **Projects → New simulation project / New test execution project** in the navigation bar.
2. Enter required fields:
   › Project name
   › Description (optional)
3. Select one model with double click.
4. Click **Create**.
   ⇨ Project is created and displayed on top of **Simulation projects overview / Test execution projects overview**.

5.11.18 Download Files of a Model

Quick Access:

Models *(navigation bar)* › *Overview* › › *Files tab* ›

1. Go to **Models → Overview** in the navigation bar.
   ⇨ All models are displayed.
2. Select model you want to open by clicking on model name or using .
   ⇨ Model details are opened.
3. Go to **Files** tab.
4. Click for download.

5.12 Report Templates

A Report template contains the following items.

- Template details: show and edit template information
- Files: show details and download a ZIP file with
  - config_signals.csv containing mapping information
  - configuration files (*.eatb and *.m)

  Reason for unavailability is shown.

Every item can be addressed via a separate tab in an opened report template.

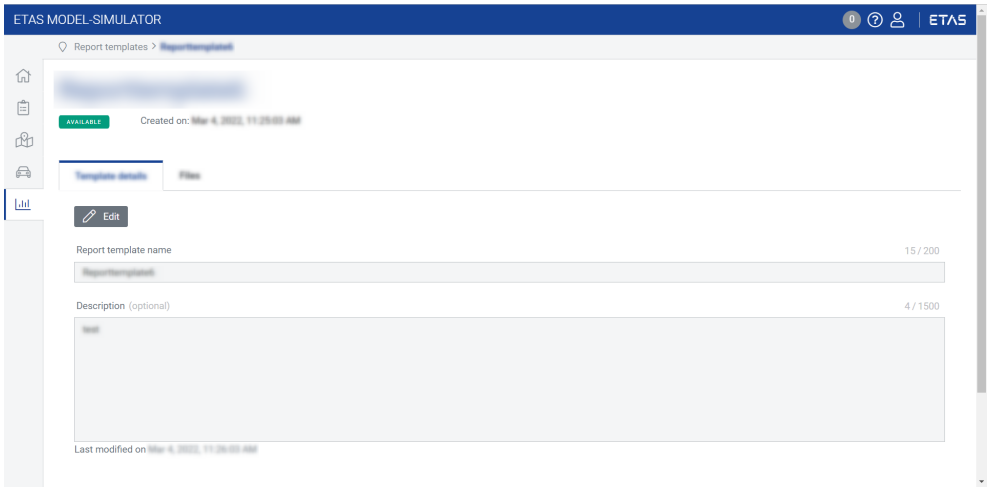
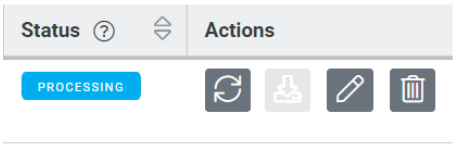
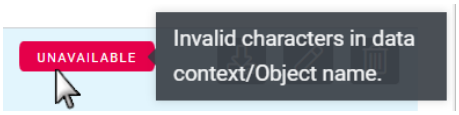


Fig. 5-10: Report template details (screenshot)

If you are interested in conceptual background information, check 3.8 "Report Templates" on page 26.

5.12.1 Status Labels for Report Templates

The overview and opened report template show the status:

Label	Description
AVAILABLE	Report template is created successfully and ready to use in simulations.
PROCESSING	Report template is created, but some process in the background is running (e.g. unzipping of files). Refresh and check the latest status, using the refresh button in the Report templates overview. 
UNAVAILABLE	Report template is created, but some background processing failed (e.g. unzipping). Report template is not available and cannot be used for reports. See more information in tooltips (hover over labels) in Report templates overview or Files tab of a report template. Check the reason for unavailability and create a new report template. 

Tab. 5-3: Overview status labels for report templates

5.12.2 Open Report Templates Overview

Quick Access:

Report templates *(navigation bar)* › *Overview*

1. Go to **Report templates** → **Overview** in the navigation bar.
   ⇨ All report templates are displayed.
2. Open report template by clicking on name or using to get to the detail site.

5.12.3 Search Report Templates

Quick Access:

Report templates *(navigation bar)* › *Overview* › Filter report templates

1. Go to **Report templates** → **Overview** in the navigation bar.
   ⇨ All report templates are displayed.

2. Type in your keyword into the search box (`Filter report templates`) to filter list of all Report templates.
   ⇨ All columns are searched while typing and results are displayed. Search operators like "", *, AND, etc. cannot be used. You can also search for creation date like "Mar" for march or for status labels like "available".

5.12.4 Sort Report Templates Overview

Quick Access:

Report templates *(navigation bar)* › *Overview* ›

1. Go to **Report templates** → **Overview** in the navigation bar.
   ⇨ All report templates are displayed.
2. Click on column heading to sort report templates ascending and descending alphabetically or by time (column `created on`).
   ⇨ Selected columns are sorted. When sorting, only the selected column is considered, the other columns are neglected.

5.12.5 Create a Report Template

Quick Access:

Report templates *(navigation bar)* › *New report template*

✓ Prerequisite: report template created. (See Required Data Structure of Report Templates)

1. Go to **Report templates** → **New report template** in the navigation bar.
2. Enter required fields:
   › Report template name
   › Description (optional)
3. Click **Upload** and select ZIP file.

   Note

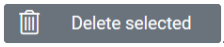
   The status check and status display VALID or INVALID refers only to the file format not to its content.

4. Click **Create**.
   ⇨ Report template is created and displayed on top of Report templates overview. Upload progress of files can be seen in the header.

5.12.6 Delete Report Template

Quick Access:

Report templates *(navigation bar)* › *Overview* ›

1. Go to **Report templates →Overview** in the navigation bar.
   ⇨ All report templates are displayed.
2. Click of the report template you want to delete (`Actions` column) and confirm dialog.
   You can also select and delete multiple report templates using the check-box and 

   .

> **Note**
>
> If you delete an item, this has an influence to the joint use and is deleted for all current users.

5.12.7 Use or Change Report Template in a Simulation Project

Quick Access:
Projects *(navigation bar)* › *Overview* › › *Project details tab* ›

✓ Prerequisite: project created. (See Create a Simulation Project)
✓ Prerequisite: report template created and labeled with AVAILABLE. (See Create a Report Template)

1. Go to **Projects → Overview** in the navigation bar.
   ⇨ All simulation projects are displayed.
2. Open simulation project by clicking on name or using to get to the detail site.
3. Go to the **Project details** tab.
4. Select **Default report template** out of the list.
5. Click **Save**.
   ⇨ The selected report template is used for upcoming automatically created reports.
   You can also create a new report with a specific template manually, see "Create a Report" on page 74.

5.12.8 Download Report Templates Files

Quick Access:
Report templates *(navigation bar)* › *Overview* › › *Files tab* ›

1. Go to **Report templates → Overview** in the navigation bar.
   ⇨ All report templates are displayed.
2. Open report template by clicking on name or using .

⇨ Template details are opened.

3. Go to **Files** tab.
4. Click for download.

6 Troubleshooting

VIRTUAL VEHICLE MODEL vs. SUBSYSTEM MODEL

What is the difference between a virtual vehicle model and a subsystem model?

A virtual vehicle model represents a powertrain system and hence expects the specific inputs vehicle target speed, gear and road slope. A subsystem model can represent any kind of system.

Read more "Models" on page 21 and "Compatibility of Campaigns and Models" on page 24.

MULTI-FACTOR AUTHENTICATION

What is a multi-factor authentication? Why do I need that?

To ensure a secure sign in into the MODEL-SIMULATOR we use a multi-factor authentication concept. This means only you can sign in with your username, password and a code received by an authentication application on your mobile device or computer.

Read more "Multi-Factor Authentication" on page 63.

PASSWORD REQUIREMENTS

Why does my password need to have 20 characters?

In general a password is better the longer it is. A password of 20 characters or more is considered to be secure.

More information "Change Password" on page 65.

EMPTY REPORT

Why is my report of the simulation project empty?

The report will be empty, if the EATB file does not fit to the model outputs. Please ensure a correct report template for the used model in your simulation project.

7 Contact Information

Technical Support

You can contact ETAS MODEL-SIMULATOR Service Desk:
ETAS MODEL-SIMULATOR Sevice Project

For details of your local sales office as well as your local technical support team and product hotlines, take a look at the ETAS website:

www.etas.com/hotlines

ETAS Headquarters

ETAS GmbH

Borsigstraße 24	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	Internet:	www.etas.com

Glossary

This glossary contains abbreviations and explanations of special terms that are important.

Campaign

A campaign is based on one or many stimuli files, which are used to feed the inputs of the simulation model. In the MODEL-SIMULATOR 3 types are selectable: subsystem campaign, virtual vehicle campaign and test execution campaign.
See 3.4 "Campaigns" on page 21.

CDFX

Calibration Data Format (ASAM standard), created with e.g. COSYM, EE (Experiment Environment) or other ASAM CDF supported tool and used to parameterize the models. It can be part of a virtual vehicle campaign or of a model (configuration files) and must be created and added manually.
See 3.4 "Campaigns" on page 21 and 5.11 "Models" on page 96.

It stores calibration parameters of different data types, the physical values and unit. The XML-based format can be easily validated, edited, imported and exported by calibration tools and XML editors.

DLC4 (ETAS specific file format)

Datalogger configuration, created with e.g. EE (Experiment Environment), which is used to limit signals for data recording. The file can be used to reduce processing time, only the listed signals will be part of the simulation result instead of all. It can be part of a model (configuration files) and must be created and added manually.
See 3.5 "Models" on page 21 and 3.9 "Upload Files Overview and Supported File Formats (Artifacts)" on page 27.

Home/dashboard

The dashboard is the starting point in MODEL-SIMULATOR and gives an overview of latest simulation projects, campaigns and models. You can create simulation projects and campaigns or upload models.

MF4/MDF

Measurement Data Format (ASAM standard) used for time series data, e.g. simulation data results or input stimuli. It is part of a campaign.
See 3.4 "Campaigns" on page 21.

Model

A model is a mathematical representation of a physical system.

It calculates the resulting system behavior (model outputs) according to the fed excitation signals (input stimuli). In the MODEL-SIMULATOR two types of model can be uploaded: subsystem model and virtual vehicle model.
While a subsystem model can represent any kind of system, a virtual vehicle model represents a powertrain system, hence expects the specific inputs vehicle target speed, gear and road slope.

Model repository

The model repository is a dedicated space to upload and manage all models in the MODEL-SIMULATOR.

Real time factor (RTF)

The RTF is a factor that describes how fast a simulation runs in comparison to real time. An initial value can be determined by running a simulation in COSYM and dividing the simulation time by the duration of stimuli file. **RTF = Duration of simulation/Duration of stimuli file**
See 3.5.1 "Real Time Factor (RTF)" on page 22.

Simulation group

A simulation group is created in the simulation project and contains the simulations to be run. A simulation group contains simulation jobs with simulation runs. A campaign is represented in a simulation job, whereas the data of a campaign (MF4 files or CDFX files) is represented in the simulation runs.
See 3.3.2 "Simulations" on page 15.

Simulation job

A simulation job contains the campaigns to be simulated, which in turn contain simulation runs.
See 3.3.2 "Simulations" on page 15.

Simulation project

A Simulation project is used to perform simulations of interest. The simulation project is based on a dedicated simulation model and suited test scenarios (campaigns) represented by simulation groups. The simulation project generates reports after a completed simulation of simulation groups.
See 3.3.1 "Simulation Projects" on page 14.

Simulation result

Output of performed simulation in form of a MF4 file.
See 3.3.2 "Simulations" on page 15.

Simulation run

A simulation run is the smallest unit within the simulation group and therefore within a simulation. It contains a trip or an arbitrary measurement file, which provides the excitation signals for the model.
See 3.3.2 "Simulations" on page 15.

SMF (ETAS specific file format)

Signal mapping file, created with e.g. Sut Mapping File Editor (usually comes with COSYM) and used for global mapping of vehicle model variables to stimuli file signal names. It can be part of a model (configuration files) and must be created and added manually.
See 3.5 "Models" on page 21 and 3.9 "Upload Files Overview and Supported File Formats (Artifacts)" on page 27.

The XML-based format can be edited by an XML editor.

STI

Generic Simulator Interface (ASAM standard), Interpolation file, created with e.g. EE (Experiment Environment) and used to interpolate input signals. It must be part of a model (configuration files) and must be created and added manually.
See 3.5 "Models" on page 21 and 3.9 "Upload Files Overview and Supported File Formats (Artifacts)" on page 27.

Stimuli files

Stimuli files are needed to feed the model inputs and thus needed to start a simulation. In the MODEL-SIMULATOR it can be measurement files as *.mf4 (subsystem campaign) or the combination of measurement files as *.mf4 and parameter files as *.cdfx in folders within a ZIP file (virtual vehicle campaign).
See 3.4 "Campaigns" on page 21.

Subsystem campaign

A subsystem campaign is a campaign based on one or many measurement files (*.mf4).
See 3.4 "Campaigns" on page 21 and 3.9 "Upload Files Overview and Supported File Formats (Artifacts)" on page 27.

Subsystem model

A subsystem model represents a part of the vehicle in form of a simulation model and can be used to represent any kind of system.
See 3.5 "Models" on page 21.

Test execution campaign

A test execution campaign is a campaign based on configured testunits created with ECU-Test. These tests are defined by packages/projects, configurations, data bases, utilities, test reports and configuration files handled in a workspace.
See 3.4 "Campaigns" on page 21 and 3.9.4 "Required Data Structure of Virtual vehicle campaigns" on page 29.

Virtual vehicle campaign

A virtual vehicle campaign is a campaign based on one or many trips (simulation runs). These trips are defined by one stimuli file (*.mf4) and one parameter file (*.cdfx), whereas the stimuli files have to contain vehicle target speed, gear and road slope.
See 3.4 "Campaigns" on page 21 and 3.9.4 "Required Data Structure of Virtual vehicle campaigns" on page 29.

Virtual vehicle model

A virtual vehicle model can be used to represent a whole vehicle in form of a simulation model with respect to a development domain like powertrain.
See 3.5 "Models" on page 21.

Figures

Fig. 3-1: Overview of MODEL-SIMULATOR elements and interaction (simulation project) 13
Fig. 3-2: Overview workflow steps in the MODEL-SIMULATOR 13
Fig. 3-3: Simulation project creation in MODEL-SIMULATOR and the containing data and outputs 14
Fig. 3-4: Structure of simulation hierarchy 16
Fig. 3-5: Shown length in files tab of a campaign 17
Fig. 3-6: Shown predicted costs and predicted time in simulations tab of a simulation project 17
Fig. 3-7: Structure of simulation hierarchy 19
Fig. 3-8: Shown estimated time and estimated costs (tests tab of a test execution project) 20
Fig. 3-9: Compatibility of inputs (models) and signals (campaigns) 24
Fig. 3-10: Use case column in the model repository 25
Fig. 3-11: Virtual vehicle campaign ZIP file composition (required structure) 30
Fig. 3-12: Content of campaign folder in virtual vehicle campaign ZIP file 30
Fig. 3-13: Content of simulation run folder in virtual vehicle campaign 31
Fig. 3-14: Test execution campaign ZIP file composition (required structure) for campaigns without libraries 32
Fig. 3-15: Test execution campaign ZIP file composition (required structure) for campaigns with libraries 33
Fig. 3-16: Model ZIP file composition (required structure) 34
Fig. 3-17: Content of model ZIP file 35
Fig. 3-18: Content of COSYMprj folder in model ZIP file (use case independent) 35
Fig. 3-19: Content of simConfig folder in COSYMprj folder 35
Fig. 3-20: Content of model ZIP file 40
Fig. 3-21: Content of COSYMprj folder in model ZIP file (test execution use case) 40
Fig. 4-1: First steps in the MODEL-SIMULATOR 53
Fig. 4-2: Model ZIP file composition (required structure) 55
Fig. 4-3: Virtual vehicle campaign ZIP file composition (required structure) 56
Fig. 4-4: Test execution campaign ZIP file composition for campaigns without libraries (required structure) 57
Fig. 4-5: Test execution campaign ZIP file composition (required structure) for campaigns with libraries 58
Fig. 5-1: Overview of user interface areas 62
Fig. 5-2: Overview of header areas 64

Fig. 5-3: Overview of dashboard (home) areas 66
Fig. 5-4: Simulation project details (screenshot) 67
Fig. 5-5: Test execution project details (screenshot) 77
Fig. 5-6: Campaign details (screenshot) 85
Fig. 5-7: Model details (screenshot) 96
Fig. 5-8: Workflow required preprocessing steps for Models 99
Fig. 5-9: Overview of steps in COSYM for model creation 100
Fig. 5-10: Report template details (screenshot) 106

Tables

Tab. 3-1: Upload files overview for MODEL-SIMULATOR 28
Tab. 3-2: Model ZIP file: content of simConfig folder and specifications 38
Tab. 3-3: Overview status labels for report status in simulation projects 42
Tab. 3-4: Overview status labels for simulation status in simulation projects 43
Tab. 3-5: Overview status labels for campaign/model status 45
Tab. 3-6: Overview tooltips for model status 49
Tab. 5-1: Overview status labels for campaigns 86
Tab. 5-2: Overview status labels for models 97
Tab. 5-3: Overview status labels for report templates 107