

ETAS MODEL-SIMULATOR V2.2

Command Line Interface



User Guide

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 V2.2 - User Guide R01 EN - 02.2024

Content

1	Safety and Privacy Information	5
1.1	Intended Use	5
1.2	Target Group	5
1.3	Privacy Notice	5
1.4	Technical and Organizational Measures	6
1.5	Safety Advice	6
2	About MODEL-SIMULATOR Command Line Interface (CLI)	7
2.1	MODEL-SIMULATOR CLI Release	7
2.2	Supported Use Case	9
2.3	Use Case Overview	9
2.4	MODEL-SIMULATOR CLI Setup	9
2.5	MODEL-SIMULATOR CLI Command Reference	10
2.6	MODEL-SIMULATOR CLI Infrastructure Setup	10
2.7	ECU-TEST Specifications (Workspace)	10
3	Setting up the MODEL-SIMULATOR CLI on a Local Machine	13
3.1	Prerequisites	13
3.1.1	General	13
3.1.2	Setting up ANSI Colors for Windows Shells	14
3.1.3	Setting up the Environment Properties for the CLI	14
3.1.4	Access	15
3.1.4.1	CLI Authentication	15
3.1.4.2	Running the CLI on the Local Machine	16
3.1.4.3	Authenticating in the CLI	16
3.1.4.4	Verifying the Upload and Download of Artifacts	17
4	MODEL-SIMULATOR Command Line Reference	18
4.1	Commands	19
4.1.1	Error handling	19
4.1.2	General Explanations	19
4.1.3	General Conventions for Uploaded Data	19
4.1.4	Help	21
4.1.4.1	Description	21
4.1.4.2	Syntax	21
4.1.5	Version	21
4.1.5.1	Description	21
4.1.5.2	Syntax	21

4.1.6	Test Run	22
4.1.6.1	Description	22
4.1.6.2	Syntax	22
4.1.6.3	Options	22
4.1.6.4	Examples	22
4.1.6.5	Behavior of Run Command	22
4.1.6.6	Input YAML File Details	23
4.1.7	Test Status	25
4.1.7.1	Description	25
4.1.7.2	Syntax	25
4.1.7.3	Options	25
4.1.7.4	Examples	25
4.1.8	Test Download	26
4.1.8.1	Description	26
4.1.8.2	Syntax	26
4.1.8.3	Options	26
4.1.8.4	Examples	26
4.1.8.5	Behavior of Download Command	27
4.1.8.6	Timeout Parameters	27
4.2	Additional Features	28
4.2.1	Consent for Safety Advice	28
4.2.2	Suppression of Safety Advice	28
4.2.3	Sample Configuration	29
4.2.4	Sample Output of the Safety Advice	30
4.2.5	Changing logLevel and Logfile Directory	31
4.2.6	Saving the Command Output in an External File	31
4.2.6.1	Description	31
4.2.6.2	Sample Configuration File	34
4.2.6.3	Output File Details	34
4.2.7	Making a Directory Case-Sensitive In Windows	35
4.2.7.1	Prerequisites	35
4.2.7.2	Check If Case-Sensitivity is Enabled	35
4.2.7.3	Enable Case-Sensitivity of an Existing Directory	35
4.2.7.4	Disable Case-Sensitivity of an Existing Directory	36
5	Error Scenarios	37
6	FAQ	48
6.1	Why is My Performance Low?	48
6.2	What Is the Difference Between consoleLogLevel and logLevel	48
7	Contact Information	49

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 ETAS 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 of test 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

ETAS MODEL-SIMULATOR CLI allows you to run the test executions in the cloud. The CLI can be installed on the local PC or can be integrated into automation scripts. Currently only Windows is supported.

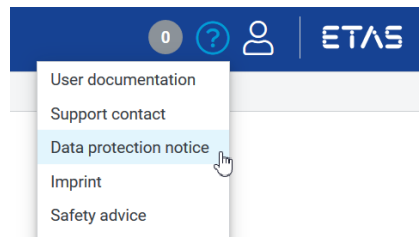
1.2 Target Group

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 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 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.

1.5 Safety Advice

Every user is obliged to read the safety advice document before using the MODEL-SIMULATOR CLI and also provide the consent as read and accepted. Every command (except help command and version command) of the MODEL-SIMULATOR CLI provides the contents of safety advice as output. You have to read the safety advice and provide consent to proceed further with the execution. For more details see "[Additional Features](#)" on page 28.

2 About MODEL-SIMULATOR Command Line Interface (CLI)

MODEL-SIMULATOR CLI allows you to run the test executions in the cloud. The MODEL-SIMULATOR CLI can be installed on the local PC or can be integrated into automation scripts.

- Only Windows is supported.
- It supports test execution with FMPy.
- Triggered test execution cannot be stopped.
- If you want to use a new campaign, you must specify a new name. Otherwise, already uploaded data with the same name will be reused.

2.1 MODEL-SIMULATOR CLI Release

The MODEL-SIMULATOR CLI is provided as a ZIP file with the following folders:

- **Documentation:** Holds all required documents.
 - **OSS Attributions:** Contains a list of open source software (OSS) components used within the product under the terms of the respective licenses. The source code corresponding to the open source components is also provided along with the product wherever mandated by the respective OSS license.
- **MODEL-SIMULATOR- <version >:** Holds the executable MODEL-SIMULATOR CLI file.
- **Templates:** Holds the reference template files.

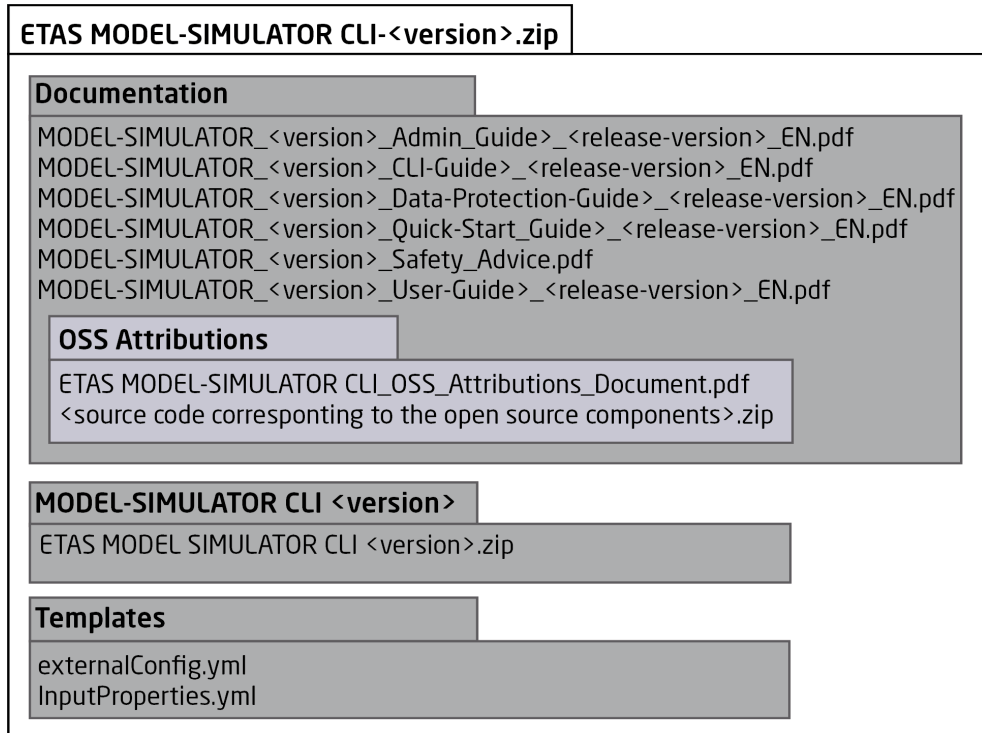


Fig. 2-1: MODEL-SIMULATOR CLI folder structure

- **<version>**: Describes the current MODEL-SIMULATOR or MODEL-SIMULATOR CLI version.
- **<release_version>**: Describes the release version of a document.

2.2 Supported Use Case

The use of MODEL-SIMULATOR is determined based on the campaign provided.

ECU-TEST is a software for testing and validation of embedded systems. MODEL-SIMULATOR accepts ECU-TEST workspaces which comprises of the test suite and the integrated System-Under-Test such as a Functional Mock-up Units (FMUs).

FMPy is an inbuilt library in ECU-TEST used to simulate the FMUs. In addition, VLAB virtual hardware models can also be used for simulation within the ECU-TEST workspace.

MODEL-SIMULATOR also supports the usage of MOBI (Modular Bus Interface) tool in the cloud. With this, the project specific SPI communication can be translated into readable instructions.

Test Execution

- **Simulator:** FMPy (provided with ECU-Test) or VLAB (V2.8.3)
- **Test automation tool:** ECU-TEST V2023.2.3 from Tracetronic
- **Interface mapping tool:** MOBI (V5.0.1)

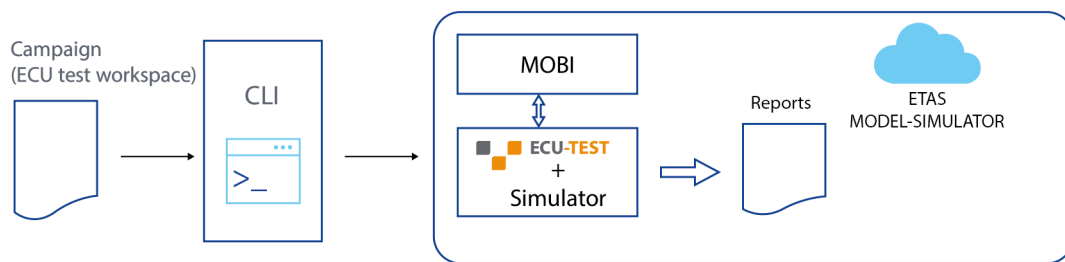


Fig. 2-2: General structure of Use Case

2.3 Use Case Overview

You can use the MODEL-SIMULATOR CLI to perform the following actions related to test execution:

- Run test
- Get status
- Download result

2.4 MODEL-SIMULATOR CLI Setup

To get more information about MODEL-SIMULATOR CLI Setup, see ["Setting up the MODEL-SIMULATOR CLI on a Local Machine"](#) on page 13.

2.5 MODEL-SIMULATOR CLI Command Reference

To get more information about MODEL-SIMULATOR CLI Command Reference, see "[MODEL-SIMULATOR Command Line Reference](#)" on page 18.

2.6 MODEL-SIMULATOR CLI Infrastructure Setup

- ETAS MODEL-SIMULATOR Service Desk must create a technical user. You can contact ETAS MODEL-SIMULATOR Service Desk using this [link](#).
- Once the environment is available, you need to update the `externalConfig.yml` with the environment access data provided by ETAS MODEL-SIMULATOR Service Desk
(You can find the template at `ETAS MODEL-SIMULATOR CLI <version> > Templates > externalConfig.yml`).

2.7 ECU-TEST Specifications (Workspace)

The ECU test tool 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.

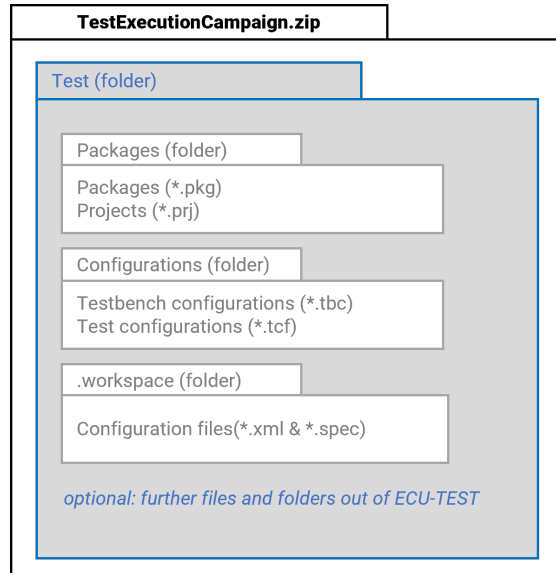


Fig. 2-3: Test execution campaign ZIP file composition (required structure) for campaigns without libraries

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.

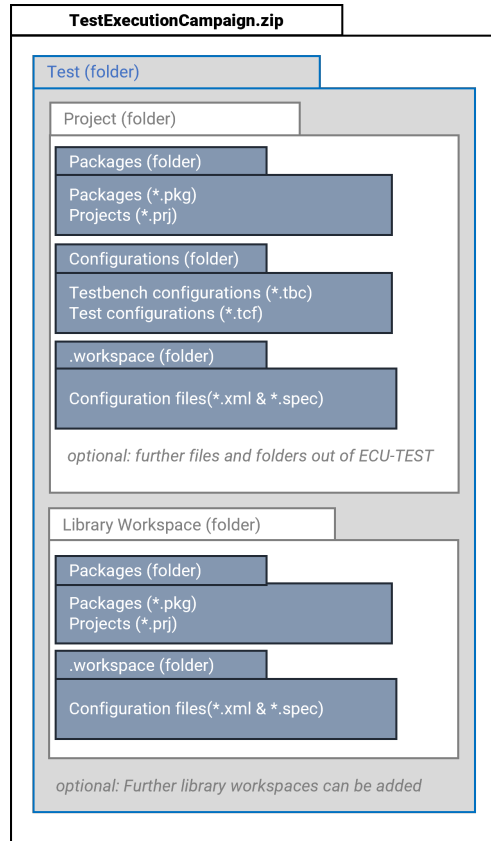


Fig. 2-4: Test execution campaign ZIP file composition (required structure) for campaigns with libraries

3 Setting up the MODEL-SIMULATOR CLI on a Local Machine

3.1 Prerequisites

3.1.1 General

- Java (JDK 11) is installed on your PC.
If Java (JDK 11) is not installed, go to [Java installation](#) and download OpenJDK11. You can check the Java version on your system by executing "java --version" in the command prompt.
- Proxy configurations are set correctly.

Import of Bosch Root CA Certificate

Execute the following command to import the certificate as trusted root CA certificate:

Adding to default Java Keystore

```
#!/bin/bash
keytool -import -trustcacerts -cacerts -storepass changeit -
alias Bosch-CA-DE -file/C/Temp/BOSCHCA-DE_bin.cer
```

Adding to non-default Java Keystore

Note

If you want to add the certificate to a Java Keystore that is not the system's default, you need to execute the command from the `lib /security` folder of the Java installation. Mind the difference.

```
#!/bin/bash
./keytool.exe -import -trustcacerts -keystore cacerts -store-
pass changeit -alias Bosch-CA-DE -file
/C/Temp/BOSCH-CA-DE_bin.cer
```

3.1.2 Setting up ANSI Colors for Windows Shells

To display the color highlights in the command prompt

1. Download the ZIP file from [here](#) and extract it to a permanent location.
2. Execute "ansicon.exe -i" from the appropriate directory for your system (x86/x64).
3. Alternatively, place the ansicon executable in your PATH or add its location to your PATH in the system variables.

3.1.3 Setting up the Environment Properties for the CLI

To set up the environment properties for the CLI

1. Create a folder named ".ecs" under the user.home directory
2. Copy the externalConfig.yml file from the Templates folder of the CLI delivery into the created folder.
3. Enter the values of the properties mentioned in the YAML file that are required for using the MODEL-SIMULATOR CLI. Contact ETAS MODEL-SIMULATOR Service Desk for any help regarding the values to be filled using this [link](#).

3.1.4 Access

3.1.4.1 CLI Authentication

Request a technical user from ETAS MODEL-SIMULATOR Service Desk for authentication when executing commands using this [link](#).

ETAS Cloud-Services MODEL-SIMULATOR

Request access from ETAS MODEL-SIMULATOR Service Desk for ETAS Cloud-Services using this [link](#). Specify a user name, user email ID, and a reason.

Extended File Sharing (for Bosch Associates)

To get the extended file sharing access to upload and download files

1. Go to the **IdM page**.
 2. Go to the **User Self Service** tab.
 3. Select **Internet Web Access** from the table.
 4. Go to the **Search for Roles** tab on the right.
 5. Search for **Extended Internet Web Access File Sharing**.
 6. Click the **+ button** in the **Action** column and add the role to the cart.
- ⇒ The **+ button** in the **Action** column becomes a check mark.



8. Click on the **shopping cart button** on top right of the page
 9. Provide a reason and click **Submit**.
- ⇒ The request is submitted to the manager. Once the manager approves, you will get file sharing privileges.

Recommendation for the approving manager:

<https://inside-docupedia.bosch.com/confluence/display/ciafsperimetersec>

3.1.4.2 Running the CLI on the Local Machine

To run the CLI on the local machine:

1. Download the MODEL-SIMULATOR CLI-<version>.zip file from the release location to a permanent location on your local machine.
2. Unzip the MODEL-SIMULATOR CLI-<version>.zip.
3. Open a new command prompt .
4. Execute the following command to set the Java and CLI path:


```
"set path=<local_path>\jdk-version\bin;<local_path>\<-
folder_name_of_extracted_cli_zip>\bin"
```

3.1.4.3 Authenticating in the CLI

To authenticate in the CLI you have the following two options:

- **Environmental Variables:** Configure the user credentials (technical user credentials) as environment variables (as shown below) so that they are not requested during command execution.

ECS_CLI_PASSWORD	ModelSimulator123456
ECS_CLI_USERNAME	tech_fmpty

Note

This configuration is not recommended for local setup.

- **Command prompt:** Specify the technical user credentials when you run the command. The prompt asks for the user credentials if they are not provided by environmental variables. The prompt will look like the following:

```
C:\repository\cli-etas-cloud-services\bin>ecs test download -p C:/Temp FMPYTESTDEMOQ204.1
Username: tech_fmpty
Password:
User login: Successful
```

In exchange for these credentials, the CLI receives a refresh token from authenticator service, which is cached. It uses this cached data for further command executions until it expires. After expiration or the cached authentication credentials are inappropriate for the provided environment, you get asked to re-enter the credentials. In most cases this happens due to environment configuration values that were changed after caching the authentication credentials of the previously configured environment.

3.1.4.4 Verifying the Upload and Download of Artifacts

To verify the upload and download of artifacts

1. Log in to the **GUI** if it is enabled for the environment.
2. The campaign with the given name must be listed there.
3. The test execution project overview must list the project with the given name and the appended sequence number.

4 MODEL-SIMULATOR Command Line Reference

The following list provides the available commands and features in the MODEL-SIMULATOR CLI.

Commands:

- "Help" on page 21
- "Version" on page 21
- "Test Run" on page 22
- "Test Status" on page 25
- "Test Download" on page 26

Additional Features

- "Consent for Safety Advice" on page 28
- "Changing logLevel and Logfile Directory" on page 31
- "Saving the Command Output in an External File" on page 31
- "Making a Directory Case-Sensitive In Windows" on page 35

4.1 Commands

4.1.1 Error handling

The error handling mechanism in the MODEL-SIMULATOR CLI is designed to handle an erroneous situation by prompting the user with a clear error message that has the following parts.

1. Which action requested by the user has failed.
2. Conveying the user the probable reason of why the error occurred.
3. What action should be taken to overcome the error.

The error handling mechanism detects and informs the user of any missing inputs that need to be provided in the configuration yaml files. Further, any errors with respect to command execution or issues with respect to the cloud infrastructure will also be conveyed to the user.

The error message will be displayed along with a suitable error code on the command prompt and logged in the corresponding logs as well. The user can utilize the error code to reach out to the Operations team for further assistance. Additionally, the exit code for errors will be provided as 2 and for successful executions it will be provided as 0.

4.1.2 General Explanations

Syntax

The general syntax uses the following expressions:

- **ecs**: Name of the tool
- **test**: Test execution workflow

Release Version

If the current released MODEL-SIMULATOR CLI is a pre-release version, each output of the command is preceded with the text "Prerelease version of ETAS MODEL-SIMULATOR CLI."

```
Prerelease version of ETAS MODEL-SIMULATOR CLI
```

InputProperties.yaml

`InputProperties.yaml` is the default yaml file and delivered to you. As a reason it is used as default for `Input file path`. You can use any other file, provided that it must be a yaml file.

4.1.3 General Conventions for Uploaded Data

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

- 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

4.1.4 Help

4.1.4.1 Description

Prints the help text. It is available for each command.

4.1.4.2 Syntax

It is possible, to use the help command as it self.

help command
<code>ecs --help</code>
<code>ecs -h</code>

It is also possible to use the help command to get help regarding a command or a sub-command.

with command
<code>ecs <command> -h</code>
<code>ecs <command> <subcommand> -h</code>

4.1.5 Version

4.1.5.1 Description

Prints the version of the MODEL-SIMULATOR CLI.

4.1.5.2 Syntax

<code>ecs --version</code>
<code>ecs -V</code>

4.1.6 Test Run

4.1.6.1 Description

Triggers the test execution for the given input configuration.

4.1.6.2 Syntax

```
ecs test run [ --nowait ] [Input file path]
```

- **run**: run/execute the tests

4.1.6.3 Options

Name	Required	Description
--nowait, -nw	No	Starts the test execution and returns the control (do not wait for execution to complete). This is a flag.
--wait	No	Starts the test execution and returns the control (waits for execution to complete). This is a flag.
Input file path	Yes	The input YAML file path. Both *.yaml and *.yml extensions are supported.

4.1.6.4 Examples

- `ecs test run -nw C:/CLI/InputProperties.yml`
- `ecs test run --nowait C:/CLI/InputProperties.yml`
- `ecs test run <c:/temp/InputProperties.yml>`

4.1.6.5 Behavior of Run Command

wait-mode

In this mode, the test execution is triggered and waits until the test execution is completed. Post completion, the results are downloaded if available and then the command execution is completed.. For this mode, the download path must be compulsorily present in `inputproperties.yml`.

**Note**

The download of the results is possible even for a test execution group that has a mixture of completed and failed runs. When a download is triggered on such a test execution group, the partial results are downloaded.

no wait mode

In this mode, the test execution is triggered and command execution gets completed. Basically the command does not wait for the test execution to complete and in case any results need to be downloaded later, a separate command needs to be triggered.

4.1.6.6 Input YAML File Details

The file has three sections:

- inputProperties
- outputProperties
- cliConfig

inputProperties

Contains the attributes that specify where the data is taken from into MODEL-SIMULATOR.

- **campaignPath**: Path of the ECU test workspace in zipped format.
- **vehiclePath**: Path of the vehicle configurations in zipped format.
- **name**: Name that applies to the current execution.

Following conventions have to be fulfilled for `name`:

Maximum characters: 200

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
- white spaces

- The campaign is uploaded with the given name. After uploading, it will be used again for the next execution if the same name is specified.
 - The project uses the same name to which the sequence number is appended. For example, if the name in the file is `TEST_CASE_FMPY`, then the project name for the first run is `TEST_CASE_FMPY.1`. This number is incremented continuously for the same name.
 - The name of the test group is the same as that of the project.
- **workflow**: Value of the workflow that must be specified.
- **Supported value**: `TEST_ECUTEST_FMPY`

Properties	Mandatory	Description
vehiclePath	No	Mandatory for test execution with COSYM only.
workflow	Yes	Always required irrespective of the type of workflow.
name	Yes	Always required irrespective of the workflow.
campaignPath	Yes	Always required irrespective of the workflow.

outputProperties

Path where MODEL-SIMULATOR writes the data. It is only applicable if the run is triggered without `nowait` flag.

- **downloadPath**: Path to which the reports on the completed test execution are downloaded.

cliConfig

Configuration for the MODEL-SIMULATOR CLI workflow (optional field). The default value is `INFO`.

- **consoleLogLevel**: Depending on the log level set, you get the information in the command prompt. Possible values are:
 - ERROR
 - FAILED
 - INFO
 - DEBUG
 - TRACE
 This applies only to the current command.

4.1.7 Test Status

4.1.7.1 Description

Gets the status of the specified test execution reference name.



Note

It is assumed that the project name and the name of the text execution group are identical.

4.1.7.2 Syntax

```
ecs test status [ --format | -f ] [ <format value> ] [<reference_name>]
```

- **status**: get the status of the group



Note

This command returns only the group level execution status, i.e. `RUNNING\PROCESSING\COMPLETED\ERROR` if the format is not specified.

4.1.7.3 Options

Name	Required	Description
--format -f	No	Specifies the format in which the output is to be printed. Type: String Supported values: JSON, TABLE
reference_name	Yes	Name of the test execution whose status is to be displayed.

4.1.7.4 Examples

- `ecs test status -f JSON TEST_CASE_FMPY.1`
- `ecs test status TEST_CASE_FMPY.1`
- `ecs test status --format TEST_CASE_FMPY.1>`

4.1.8 Test Download

4.1.8.1 Description

Downloads the results of the completed test execution to the specified path.



Note

It is assumed that the project name and the name of the text execution group are identical.

4.1.8.2 Syntax

```
ecs test download [ < --downloadPath | -p > ] [ < Path > ]
[<reference_name>]
```

- **download**: download the results

4.1.8.3 Options

Name	Required	Description
--downloadPath -p	Yes	Location where the report must be downloaded
reference_name	Yes	Name of the test execution whose results are to be downloaded.

4.1.8.4 Examples

- `ecs test download -p C:\\temp`
- `ecs test download --downloadPath C:\\temp`

4.1.8.5 Behavior of Download Command

The `download` command downloads the test execution results for a specific project reference provided the results are available. By default, the download command waits until the archiving of the results are completed. Additionally, a specific time can be configured to wait for the download to complete. In case, the duration to download exceeds the configured limit then the download command terminates.



Note

The download of the results is possible even for a test execution group that has a mixture of completed and failed runs. When a download is triggered on such a test execution group, the partial results are downloaded.

4.1.8.6 Timeout Parameters

There are timeout parameters for the download command:

- `maxRequests` (optional): This is the number of counts that CLI needs to check with ECS cloud whether the archive of the requested test results are completed. It is not mandatory to specify this number. If this number is not specified, the check will be performed until the download is completed.
- `requestFrequency` (optional): This is the wait time between each check of whether the process was completed.
 - If the `requestFrequency` value is less than 1000, the default value of `requestFrequency` will be set as 1000.
 - If there is no value set for `requestFrequency`, then the default value for `requestFrequency` is 5000.

```
download:
  maxRequests: 50
  requestFrequency: 5000
```

Fig. 4-1: Timeout parameters

4.2 Additional Features

4.2.1 Consent for Safety Advice

The consent for the safety advice can be provided through CLI or through `externalConfig.yml` file.

Acceptance via CLI:

- You can provide consent by typing `y`, which is case insensitive on the command prompt.

Acceptance via `externalConfig.yml`:

- You can configure the key `safetyadvice.acknowledge` in the config file with the required value `true\any value`.

4.2.2 Suppression of Safety Advice

A configuration to control the suppression of safety advice content on command prompt is provided. Suppression can be done by providing the flag `safetyadvice.suppress` value as `true` in the `externalConfig.yml` file. However, under few conditions, it won't be possible to suppress the content of safety advice on command prompt even if the value is set to `true` in the `externalConfig.yml` file.

for release versions

- at the first use of a new version of MODEL-SIMULATOR CLI

for pre-release versions

- at the first use of a new version of MODEL-SIMULATOR CLI (any first command) on a particular machine + user
- at the first use of the of MODEL-SIMULATOR CLI (any first command) on a particular machine + user on a new day



Note

If the safety advice `acknowledge` has any value other than `true` in the `externalConfig.yml`, the user is prompted to accept it via MODEL-SIMULATOR CLI.

Irrespective of safety advice content is suppressed or not, the consent has to be provided through the `externalConfig.yml` or command prompt for further execution of command.

The safety advice shall not be output to the prompt in case of the `help` and `version` commands. In case of pre-release version, only pre-release text will be output for `help` and `version` command.

The value for the config properties `safetyadvice.suppress` and `safetyadvice.acknowledge` should not start with `@` symbol.

In order to suppress or acknowledge the safety advice through `externalConfig.yml` located in the path `user.home/.ecs/`, you need to include the properties with valid values as mentioned below.

Name	Required	Description	Accepted Values
<code>safetyadvice.suppress</code>	No	Used to suppress safety advice	true = Suppress the display of safety advice on command prompt any value other than true = Do not suppress the display of safety advice on command prompt
<code>safetyadvice.acknowledge</code>	No	Used to provide consent for safety advice	true = Consent for safety advice is provided any value other than true = Do not provide the consent of safety advice

4.2.3 Sample Configuration

```

ecs:
  service:
    url:
      testExecutionProject:
      campaign:
      vehicle:
      mscontext:
  artifacts:
    storage:
      location:
      vehicleName:
  aws:
    region:
    clientId:
    userpoolId:
    endpoint:
    identitypoolId:
  proxy:
    enabled:
    host:
    port:
  runtime:
    generic:
      maxRequests:
      requestFrequency:
    download:
      maxRequests:
      requestFrequency:
  output:
    logLevel:
    logPath:
    file:
    folderPath:
    format:
  safetyadvice:
    suppress: true
    acknowledge: true

```

Fig. 4-2: Sample Configuration

4.2.4 Sample Output of the Safety Advice

```

C:\Program Files\Etag\bin>check.exe /s:en /p:RIP /r:RIP
Please use version of ETAS MODEL-SIMULATOR 2.11
SAFETY ADVICE (english)

Warning! It is critical that you read and follow this safety advice, the product description including technical data and the associated technical documentation, which are facilitated on and
www.etas.com (via direct product access/select product). Do not use the product if you cannot read and/or understand the information for safe operation. If you do have questions for safe operation,
contact your region www.etas.com/offices.

With this ETAS product, you can influence safety-relevant system or data, e.g. in automobiles, automobile components and test benches. Improper use and unskilled application without adequate
training of such products may cause threats to life and physical conditions as well as damages to property.

Our products have been developed and released exclusively for use in applications defined in the product description.

Fitness and suitability of the products for any intended use beyond the utilization for which the products have been released (e.g. different stresses/strains or technical conditions) need to
be verified by taking appropriate actions and measures (e.g. by means of tests).

ETAS products made available as beta versions of firmware, hardware and software are to be used exclusively for testing and evaluation. These products may have not sufficient technical documenta
tion regarding quality and accuracy for market-released series products. Therefore product performance may differ from the product description and your expectations. The product should be used
only for testing and evaluation. Do not use data and results from beta versions without prior and separate verification and validation and do not pass them to third parties without prior examination.

Do not use this product if you do not have proper experience and training in using the product.

To allow proper handling of ETAS products ETAS has released known issue reports (KIR) on its website. Known Issue Reports provide information on known product problems of substantial relevance
and give instructions on available solutions. Prior to the initial operation of the product you are required to verify whether a KIR is available for the current product version and address
known issue reports can be found on ETAS website www.etas.com/KIR (password in KIR area: KIRAGIR).

Program code and/or control actions of programs, which have been developed or modified as well as data of any kind, which have been identified by using ETAS products, will need to be verified
for safety and suitability prior to any use or dissemination.

When using this product with systems which accomplish safety functions (e.g. in automobiles, automobile components and test facilities) that influence system behaviour and can affect the safety
of the system, ensure that the system can be transitioned to a safe condition (e.g. emergency shutdown or emergency operation mode) if a malfunction or hazardous incident should occur.

All applicable regulations and statutes regarding operation must be strictly followed when using this product.

Use of this ETAS product or any program code, program control procedures in the public domain (e.g. on public roads) should not occur unless they have been tested and verified as being safe
for use in the products only in closed and designated test environment.

ETAS Contact Addresses
ETAS Headquarters
ETAS GmbH
Rueschallee 24 Phone: +49 711 3423-0
70469 Stuttgart Fax: +49 711 3423-2100
Internet Mail: www.etas.com

Warning! If you fail to follow this safety advice, there might be a risk of death, serious injury or property damage. The ETAS Group and their representatives shall not be liable for any damage
of the product. ETAS provides trainings regarding the proper and intended use of this product.

template: 00010004 V17, released 2022-02-10
Finally provide the consent for safety advice to proceed further (Y/N)?Y
consent provided for safety advice through command prompt

```

Fig. 4-3: Sample Output of the Safety Advice

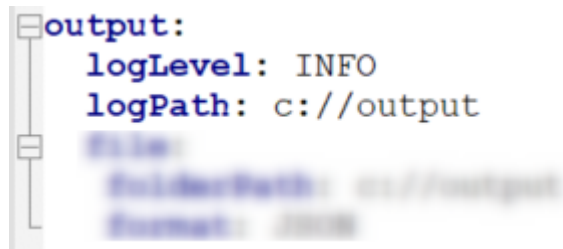
4.2.5 Changing logLevel and Logfile Directory

It is possible to change the `logLevel` and Logfile Directory by using `logPath` in output section in the `externalConfig.yml`.

- `logLevel` (optional): To change the Log level with respect to application for which the respective information will be logged in log files. If a `logPath` is not specified, then by default the log path will be created in the `.ecs` (home) directory.

Possible values for logLevel:

- OFF
 - FATAL
 - ERROR
 - INFO
 - DEBUG
 - TRACE
 - ALL
- `logPath` (optional): To change the folder location where the CLI logfile to be created.



```
output:
  logLevel: INFO
  logPath: c://output
  file:
    folderPath: c://output
    format: JSON
```

Fig. 4-4: LogLevel and logPath

4.2.6 Saving the Command Output in an External File

4.2.6.1 Description

When a command is executed in the MODEL-SIMULATOR CLI, certain information is output on the command prompt and also in the log file. The subsequent command execution sometimes requires this information as an argument. As a result you must parse the output to extract the data which is cumbersome. To enable this, the MODEL-SIMULATOR CLI provides a configuration where the information about the current command executed and the output of the same could be persisted in a json file. The configuration must be provided in the `externalConfig.yml` file as below.

Output

- `file`
 - `folderPath`: <valid folder path to store the output file >
 - `format`: JSON

Name	Required	Description	Accepted Values
output.file.folderPath	Yes (Not mandatory for the MODEL-SIMULATOR CLI functioning)	Folder location where the output json to be created.	[ValidFolderPath]
output.file.format	No	Used to provide a format in which the output file needs to be created. Currently only JSON format is supported.	JSON/[blank]

The output file name is the name of the test execution given by the user without the sequence number.

E.g., if you execute the status command by giving the reference name as `TEST.1` then the file name will be `test.json`.

All the subsequent commands executed for the same name, will be part of the `test.json` file.

When the next command is executed for the same reference name, the previously executed command info will be part of the same file in the history section.



Note

Only if the command execution is success, then information is persisted.
If an invalid folder path is provided, then MODEL-SIMULATOR CLI will provide an error message and proceed with the command execution.
The value for the config properties `output.file.folderPath` and `output.file.format` should not start with @ symbol.

When you try to execute a command with a project name that differs only by its case and its sequence number, the output of those commands gets stored in a single JSON file.

As the Windows operating system treats two file names that differ only in upper and lower case as a single file, this problem occurs, also described in [Fig. 4-5](#) and [Fig. 4-6](#)


```

C:\Users\>ecs test run -nw D:\ETAS_CLI\InputProperties.yaml
Consent provided for safety advice through external config

Input properties loaded: Successful
Username:
Password:
User login: Successful
New test execution project created with name: projectname.1
Test execution trigger: Successful

C:\Users\>ecs test run -nw D:\ETAS_CLI\InputProperties.yaml
Consent provided for safety advice through external config

Input properties loaded: Successful
Username:
Password:
User login: Successful
New test execution project created with name: PROJECTNAME.1
Test execution trigger: Successful

```

Fig. 4-5: Project names with upper and lower case

```

"commandInformation": {
  "name": "PROJECTNAME.1",
  "lastExecutedCommand": "ecs test run -nw D:\\
  "timestamp":
},
"history": [
  {
    "name": "projectname.1",
    "lastExecutedCommand": "ecs test run -nw D:\\
    "timestamp":

```

Fig. 4-6: JSON file output

To solve the problem, see "Making a Directory Case-Sensitive In Windows" on page 35.

4.2.6.2 Sample Configuration File

```

ecs:
  service:
    url:
      testExecutionProject:
        campaign:
        vehicle:
        mscontext:
artifacts:
  storage:
    location:
    vehicleName:
aws:
  region:
  clientId:
  userpoolId:
  endpoint:
  identitypoolId:
proxy:
  enabled:
  host:
  port:
runtime:
  generic:
    maxRequests:
    requestFrequency:
  download:
    maxRequests:
    requestFrequency:
output:
  logLevel: INFO
  logPath: c://output
  file:
    folderPath: c://output
    format: JSON
safetyadvice:
  suppress:
  acknowledge:

```

Fig. 4-7: Sample Configuration File

4.2.6.3 Output File Details

The file has two sections:

- commandInformation
- history

commandInformation

This section contains the output information of the recently executed command which can be used as input to the other commands.

- name
Name that applies to the current execution with sequence number which will indicate the number of retries done with the same inputs passed.
- lastExecutedCommand
Command that got triggered recently for the particular test execution.
- timestamp
Timestamp in which the mentioned command is executed for the particular test execution.
- downloadedPath as download command
Contains the test result downloaded path when the command gets executed.
- status as status command
Contains the details as below:

- `resultStatus`

Contains the result information

- `total`: The total number of test cases ran as.
- `inconclusive`: The number of test cases where the result is inconclusive.
- `success`: The number of test cases where the result is a success.
- `error`: The number of test cases where the result is an error.
- `unavailable`: The number of test cases where the result is not available

- `executionResult`

Contains the test execution status.

history

This section contains all the information of the previously executed commands, arranged in a manner where the most recently executed command information can be found first.

4.2.7 Making a Directory Case-Sensitive In Windows

4.2.7.1 Prerequisites

Changing the case-sensitivity of a directory require elevated permissions (Administrator rights). Changing the case-sensitivity flag also requires “Write attributes”, “Create files”, “Create folders” and “Delete subfolders and files” permissions on the directory. For more information, see the this [troubleshooting](#).

4.2.7.2 Check If Case-Sensitivity is Enabled

1. Execute the command "`fsutil.exe file queryCaseSensitiveInfo <path>`" in the PowerShell where `<path>` needs to be replaced with your file path.

The command output will look like the below:

```
PS U:\> fsutil.exe file queryCaseSensitiveInfo
Case sensitive attribute on directory is disabled.
```

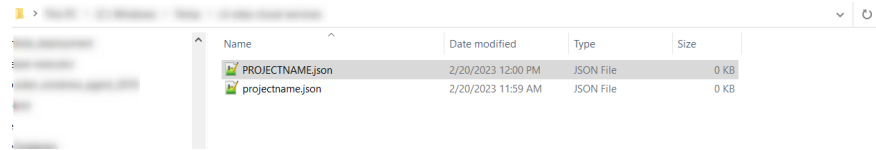
4.2.7.3 Enable Case-Sensitivity of an Existing Directory

1. To change a directory in the Windows file system so that it is case-sensitive (FOO ≠ foo), run PowerShell as Administrator.
 2. Execute the command "`Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux`".
- ⇒ After being completed, a restart is required.
3. Execute the restart.
 4. Execute the command "`fsutil.exe file setCaseSensitiveInfo <path> enable`" where `<path>` needs to be replaced with your file path.

The command output will look like the below:

```
PS C:\WINDOWS\system32> fsutil.exe file SetCaseSensitiveInfo enable
Case sensitive attribute on directory is enabled.
```

- ⇒ After executing the above command you are able to save two files with name that only differs in cases in the same directory.



Name	Date modified	Type	Size
PROJECTNAME.json	2/20/2023 12:00 PM	JSON File	0 KB
projectname.json	2/20/2023 11:59 AM	JSON File	0 KB

4.2.7.4 Disable Case-Sensitivity of an Existing Directory

1. To change a directory in the Windows file system back to the case-insensitive default (FOO = foo), run PowerShell as Administrator.
2. Execute the command "fsutil.exe file setCaseSensitiveInfo <path> disable" where <path> needs to be replaced with your file path.

- ⇒ If the requested directory has files with names that only differ in its case then you will end up in an error like below:

```
PS C:\WINDOWS\system32> fsutil.exe file setCaseSensitiveInfo C:\Temp\cli-etcs-cloud-services disable
Error: This directory contains entries whose names differ only in case.
```

3. Remove those files and try to execute the command again.
- ⇒ After executing the above command, the case-sensitivity is disabled.

```
PS C:\WINDOWS\system32> fsutil.exe file setCaseSensitiveInfo C:\Temp\cli-etcs-cloud-services disable
Case sensitive attribute on directory C:\Temp\cli-etcs-cloud-services is disabled.
```

5 Error Scenarios

When working with MODEL-SIMULATOR CLI, you may receive error codes in the CLI. An overview of possible error scenarios and how to correct them is given in the following table.

Error Code	Error Description	Scenario
1020000012	User sign-in failed due to Incorrect username or password. Try again with valid user credentials.	Provided user credentials like user name and password as an environmental/command prompt is invalid (non empty but invalid).
	The user sign-in failed as only technical user credentials are permitted for CLI authentication. Contact Operations team for relevant technical user details.	Provided non-technical user credentials, CLI supports only technical user.
	The password has to be changed before the first login. Reset the initial password in MODEL-SIMULATOR GUI and retry.	The provided technical user credentials has been used without resetting the initial password.
	User sign-in failed due to User pool client XXX does not exist. Try again with valid user credentials.	An error occurred while trying to sign in user using the login credentials provided.
1020000029	Failed to start the test execution. The upload of the campaign is unsuccessful due to an internal error. Try re-uploading the campaign.	An error occurred while trying to upload or create a campaign.
1020000030	Failed to complete the run command execution. Possible reasons: — Internal API call failed. — Server error while processing the request. Contact the Operations team.	An error occurred while trying to get the status of a project or group after a test execution was successfully triggered.

1020000036	Failed to complete the run command execution as the download of results failed. It is possible that the download path provided is not valid or some internal error occurred. Try downloading again or if the issue persists, contact Operations team.	An error occurred while trying to download the test result of the test execution.
1020000046	Failed to get the status of the test execution. The provided reference name XXX is not found. Provide the proper project/group reference.	CLI was not able to find the mentioned test execution project/group with the given reference name.
1020000051	Failed to download the result. The test execution is in PROCESSING state for the requested reference name and hence no results are available for download. Try downloading the results after sometime.	The results could not be downloaded as the execution is in progress.
	Failed to download the result. The result archive is in RESULTS_SUBMITTED_FOR_ARCHIVE state for the requested reference name. Wait until the result archive get concluded.	The results could not be downloaded as archiving of results is not completed.
	Failed to download the result. The result archive is in ARCHIVE_REQUEST_TERMINATED state for the requested reference name and hence no results are available for download. Contact the Operations team.	The results could not be downloaded as the archive status is TERMINATED.

1020000052	Failed to complete the run command execution as the download of results failed. An interruption occurred while trying to wait for the results to be archived. Try downloading after some time.	An error occurred while trying to wait until the archive of the test execution result was completed.
1020000053	Failed to complete the run command execution as the download of results failed. The time for archiving of the results exceeded the configured time limit. Try downloading the results after some time.	The archiving of the test execution result took more than the configured time.
1020000071	Failed to start the test execution. Upload of the artifact failed due to Identity Pool XXX not found. Try again with proper identity pool ID.	An error occurred while trying to upload or create a artifact as the identity pool ID configured in the external configuration file is incorrect.
1020000080	The provided reference name XXX has invalid characters. Refer to the CLI user guide for a list of allowed characters.	Reference name provided in <code>Input-properties.yaml</code> / command arguments has invalid characters. See General Conventions for Uploaded Data for a list of allowed characters.

1020000081	Failed to start the test execution. The provided campaign path does not exist or is not readable. Provide a valid campaign path.	The campaign path provided is not valid.
	Failed to start the test execution. The provided name: {name-value} has more than {max-character-limit} characters. Provide a name within {max-character-limit} characters.	The provided value for name has more than 200 characters.
	Failed to start the test execution. The provided reference name XXX has invalid characters.	The provided value for name has invalid characters. See General Conventions for Uploaded Data for a list of allowed characters.
	Failed to start the test execution. The provided campaign path does not exist or is not readable. Provide a valid campaign path.	The provided campaign path does not exist. The provided campaign path exists but not readable.
	Failed to start the test execution. Provided campaign file extension is not supported. Provide campaign file in zip format.	The provided campaign path is not in ZIP format.
	Failed to start the test execution. Download path is not provided and is mandatory when test execution is triggered in wait mode. Provide a valid download path.	The mandatory download path is not provided when test execution is triggered in wait mode.
	Failed to start the test execution. The provided download path does not exist or is not writable. Provide a valid download path.	The provided download path does not exist in the local folder path. The provided download path is not writable.

	Failed to start the test execution. The provided vehicle path does not exist or is not readable. Provide a valid vehicle path.	The provided vehicle path does not exist. The provided vehicle path exists but not readable.
	Failed to start the test execution. The provided vehicle file extension is not supported. Provide vehicle file in zip format.	The provided vehicle path is not in ZIP format.
1020000122	Unable to cache the user authentication information. Check the configFile write permissions inside ecs folder. Retry sign in and if the issue persists, contact Operations team.	Error occurred while caching the authentication information.
1020000141	Failed to start the test execution. The provided user credentials are empty. Try again with valid user credentials.	Provided user credentials like user name and password as an environmental variable/command prompt is an empty string.

1020000164	Failed to start the test execution as the uploaded campaign was in UNAVAILABLE status. The campaign is unavailable because the .workspace folder/s Packages folder/s Configurations folder/s in Root folder were not found. Refer to the user documentation for correct test campaign configuration	Corrupted campaign or missing mandatory folders within campaign ZIP file.
<hr/>		
	Failed to start the test execution. The uploaded campaign was in UNAVAILABLE status. The campaign is unavailable because the uploaded zip file was empty. Refer to the user documentation for correct test campaign configuration.	Campaign ZIP file is empty.
<hr/>		
1020000168	Failed to start the test execution. An interruption occurred while trying to get status for the Campaign. Try uploading the Campaign after some time and if the issue persists, contact Operations team.	An error occurred while trying to wait until the campaign came to AVAILABLE status.

1020000169	Failed to start the test execution. The uploaded campaign is in PROCESSING status. Increase the value of numberOfRetries attribute in the external configuration file and try to trigger test execution again.	The created campaign was not in an AVAILABLE state within the configured time, hence could not proceed further.
1020000181	Failed to start the test execution. The vehicle with name XXX was not found. Contact Operations team for more information.	CLI was not able to find the vehicle with the given vehicle name value in the external configuration file.
1020000187	Failed to start the test execution. The upload of the vehicle is unsuccessful due to an internal error. Try re-uploading the vehicle.	An error occurred while uploading vehicle.
1020000188	Failed to start the test execution. The uploaded vehicle is in PROCESSING status. Increase the value of maxRequests attribute in the external configuration file and try to trigger test execution again.	The created vehicle was not in an AVAILABLE state within the configured time, hence could not proceed further.
1020000191	Failed to start the test execution. The uploaded vehicle was in UNAVAILABLE status. The vehicle is unavailable because the mandatory deployable zip file is not found. . Refer the user documentation for more detailed information about the model configuration	Corrupted vehicle or missing mandatory folders within vehicle ZIP file. Vehicle ZIP file is empty.

1020000203	Failed to start the test execution. The provided path does not have the necessary security permissions to be read/written. Provide a valid path with the necessary permissions.	The provided path (campaign path, vehicle path, download path) in <code>inputProperties.yaml</code> does not have the necessary security permissions to be read/written.
1020000265	Failed to read/display the Safety advice. The safety advice packed with the client is either missing or corrupted. Contact the Operations team.	An error occurred while trying to read the safety advice document to display to the user.
1020000270	The requested output format: {requestedOutputFormat} is invalid, and the acknowledgement messages cannot be saved in the requested format. The supported format is JSON.	The provided output format provided in external configuration file for the history file is not supported.

1020000279	<p>Failed to start the test execution. Property key XXX should be present under XXX in externalConfig.yaml / InputProperties.yaml. Update the mentioned file/- files with the suggested changes and retry.</p>	<p>Mandatory key not provided in the external configuration file/ <code>InputProperties.yaml</code> file.</p>
	<p>Failed to start the test execution. Property value of XXX cannot be empty in externalConfig.yaml / InputProperties.yaml. Update the mentioned file/- files with the suggested changes and retry.</p>	<p>Value is not provided for mandatory key in the external configuration file/ <code>InputProperties.yaml</code> file.</p>
	<p>Failed to start the test execution. Property value of XXX cannot be empty and should be either true or false in in externalConfig.yaml / InputProperties.yaml. Update the mentioned file/files with the suggested changes and retry.</p>	<p>Empty value is provided for mandatory key in the external configuration file/ <code>InputProperties.yaml</code> file.</p>
	<p>Failed to start the test execution. External configuration file/Configuration properties file is empty. Update the mentioned file/files with the suggested changes and retry.</p>	<p>External configuration file is empty.</p>

1020000282	<p>CLI startup failed. The provided externalConfig.yaml is invalid.</p> <ul style="list-style-type: none"> – Property keys may be duplicate. – Wrong indentation/spaces around the property – Special/junk characters around property keys. <p>Make sure the file is valid by considering the suggested points and retry or contact operations team.</p>	<p>Misaligned one/multiple key in the yaml file.</p> <p>A mandatory set of key is repeated.</p> <p>Special characters provided in front of any mandatory key. E.g. '!' was added in front of Key 'artifacts'.</p> <p>Some invalid special characters are -! > }] , .</p>
1020000283	<p>Failed to start the test execution. YML/YAML processing failed due to configuration properties file being invalid.</p> <ul style="list-style-type: none"> – Property keys may be duplicate. – Wrong indentation/spaces around the property keys/values in the files. – Special/junk characters around property keys. <p>Make sure the file is valid by considering the suggested points and retry or contact operations team.</p>	<p>Special characters provided in front of any mandatory key. E.g. '!' was added in front of Key 'artifacts'.</p> <p>Some invalid special characters are -! > }] , .</p>
1020000285	<p>CLI startup failed. Contact operations team.</p>	<p>CLI fails to start because of some configuration issue.</p>
1020000290	<p>The arguments provided in the command is invalid. Check the argument or syntax and try again.</p>	<p>Invalid arguments provided during command. Double - with a space.</p> <p>Eg: (double hyphen is before nw)</p> <pre>ecs test run - nw C:\TestData\Cli_Testing\InputProperties.yml</pre>

1020000301	Valid value was not provided for safety advice. Provide a valid value.	Acknowledgment provided for safety advice with an invalid value.
1020000320	Consent to safety advice was not provided. Provide consent to proceed with command execution.	Acknowledgment provided for safety advice with an invalid value.
1020000321	<ul style="list-style-type: none"> – Internal API call failed. – Server error while processing the request. Contact the Operations team.	Any error that occurred apart from the specific once listed in this table.
1020000323	CLI startup failed. Property value of 'logpath' is not valid in externalConfig.yml. Provide a valid path.	Invalid logpath provided in the external configuration file that contains certain restricted special characters/character combinations. (E.g. add %& in the logpath).

6 FAQ

6.1 Why is My Performance Low?

If you notice a performance lag when executing MODEL-SIMULATOR CLI commands, you should increase the number of processors to the maximum in your system.

To increase the number of processors

1. Open run command by pressing **Windows + R**.
2. Execute `msconfig` to open a new window.
⇒ The **System Configuration** Window opens.
4. Navigate to **Boot** tab.
5. Click **Advanced Options**.
6. Set the value in the **Number of processors** tab to the maximum.
7. Click **OK**.
8. Click **Apply**.

6.2 What Is the Difference Between consoleLogLevel and logLevel

consoleLogLevel in InputProperties.yaml file

Depending on the log level set, you get the information directly in the command prompt, see "[cliConfig](#)" on page 24.

LogLevel in the ExternalConfig.yaml file

Depending on the log level set, you get the information in the log file, see "[Changing logLevel and Logfile Directory](#)" on page 31.

7 Contact Information

Technical Support

You can contact ETAS MODEL-SIMULATOR Service Desk:

[ETAS MODEL-SIMULATOR Service 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