LABCAR-AUTOMATION 4.2.3
User's Guide

DRIVING | EMBEDDED EXCELLENCE

Copyright

V4.2.3 R01 EN - 02.2017

Contents

1 Introduction

LABCAR-AUTOMATION is a complete software package for creating, managing and running automated ECU tests.

It makes it possible to define test cases which are not specific to one test bench thus ensuring that test cases can be reused throughout the development cycle.

The test process in LABCAR-AUTOMATION is divided into a number of functional sections. A "role" is mapped for each of these functional sections in LABCAR-AUTOMATION which works with tools corresponding to its special task.

This chapter includes information on the following topics:

- "About This Manual" on page 8

  This section describes the structure of this manual and contains tips on its use.
- "Roles in the Test Process" on page 12

  This section describes how the different tasks are assigned to the different roles in the test process in LABCAR-AUTOMATION 4.2.3.
- "LABCAR-AUTOMATION 4.2.3 Software Applications" on page 18

  This contains a description of the software applications used for the tasks of the relevant role.
- "The LABCAR-AUTOMATION 4.2.3 Test Project" on page 20

  This section describes the structure of a LABCAR-AUTOMATION 4.2.3 test project.
- "Configuration Files" on page 22

  This section contains information on the function of LABCAR-AUTOMATION configuration files.

1.1 About This Manual

This section describes the structure of this manual and contains tips on its use.

1.1.1 Structure of the Manual

This manual contains information on LABCAR-AUTOMATION 4.2.3. The individual sections are:

- "Introduction" on page 7

  This chapter.
- "Installation" on page 27

  This chapter contains a detailed description of the installation of LABCAR-AUTOMATION 4.2.3.
- "The Test Manager User Interface" on page 39

  This chapter describes the user interface of the main application within LABCAR-AUTOMATION 4.2.3.
- "Working with LABCAR-AUTOMATION 4.2.3" on page 51

  This chapter describes the tasks in a test process which are to be carried out by the different roles in the different applications of LABCAR-AUTOMATION 4.2.3.
- "Configuration Files and their Editors" on page 147

  This chapter contains a description of the configuration files which are of significance in a LABCAR-AUTOMATION project and their editors.
- "Tutorial" on page 197

  This chapter helps you to learn how to work with LABCAR-AUTOMATION 4.2.3 using a simple example.
- "Glossary" on page 237

  The Glossary contains definitions of terms which are of significance in LABCAR-AUTOMATION 4.2.3.

A list of ETAS contact addresses and an Index conclude the manual.

1.1.2 Correct Use of the Software

LABCAR-AUTOMATION is intended exclusively for the automation of sequences at Hardware-in-the-Loop test benches for vehicle ECUs and must be operated by technical experts in laboratory environments only.

ETAS provides training sessions on the correct operation of LABCAR-AUTOMATION.

1.1.3 Safety Advice

DANGER!

LABCAR-AUTOMATION provides capability to automatically control various tools (of ETAS as well as other vendors). Therefore, be always aware of the actions that you may trigger with executing tests. Assure that there is no risk of damaging test equipment or even harm people lingering in the near of the test bench. Consider the hints in the Safety Advice document (available at C:\Program Files (x86)\ETAS\LABCAR-AUTOMATION 4.2\Documentation\ ETAS_Safety_Advice.pdf and the product DVD).

1.1.4 Using This Manual

Representation of Information

All activities to be carried out by the user are shown in what we call a "Use-Case" format, i.e. the target to be achieved is defined briefly in the title and the individual steps necessary to achieve this target are then listed. The information is displayed as follows:

Target definition

Any introductory information...

- Step 1

  Possibly an explanation of step 1...
- Step 2

  Possibly an explanation of step 2...
- Step 3

  Possibly an explanation of step 3...

Any concluding remarks...

Specific example:

To create a new file

If you want to create a new file, no other file may be open.

- Select File → New.

  The "Create File" dialog box appears.
- Enter a name for the file in the "File name" box.

  The file name must not be more than 8 characters long.
- Click OK.

The new file is created and saved under the name specified. You can now work with the file.

Typographic Conventions

The following typographic conventions are used:

Select **File** → **Open**.	Menu functions are shown in boldface/blue.
Click **OK**.	Buttons are shown in boldface/blue.
Press <ENTER>.	Keyboard commands are shown in angled brackets in block capitals.

The ”Open File” dialog box appears.	Names of program windows, dialog boxes, fields etc. are shown in quotation marks.
Select the file `setup.exe`.	Text in drop-down lists, program code, as well as path and file names are shown in the `Courier` font.
A conversion between the file types logical and arithmetic is *not* possible.	Content markings and newly introduced terms are shown in *italics*

Important notes for the user are shown as follows:

Note

Important note for the user.

1.2 Roles in the Test Process

At an early stage of the test procedure, there will be only a few experts which are responsible for all procedures and tasks of the test project. With increasing complexity, detail and distribution of a test, it is a good idea to distribute the subtasks to several roles.

A first step would be to leave the development of test cases and the creation of executable test projects to a group of test experts, while, for example, others, who do not need to know every single detail of the test, deal with the execution. A further task distribution could, for example, be in the division of the development of test cases which are still of a relatively general nature and the creation of the more specific executable test projects which are already aimed at specific UuTs.

LABCAR-AUTOMATION 4.2.3 introduces a consistent distribution of the different tasks involved in the creation and execution of test processes which is described in this section. It is entirely possible and, depending on the scope and complexity of the test, sensible for several of these roles to be united in one person.

The different roles and the applications on which these roles work are shown in Fig. 1-1 on page 13.

At the same time, LABCAR-AUTOMATION 4.2.3 consists of different subapplications which are closely connected to the roles described below. These subapplications are described in the section "LABCAR-AUTOMATION 4.2.3 Software Applications" on page 18.

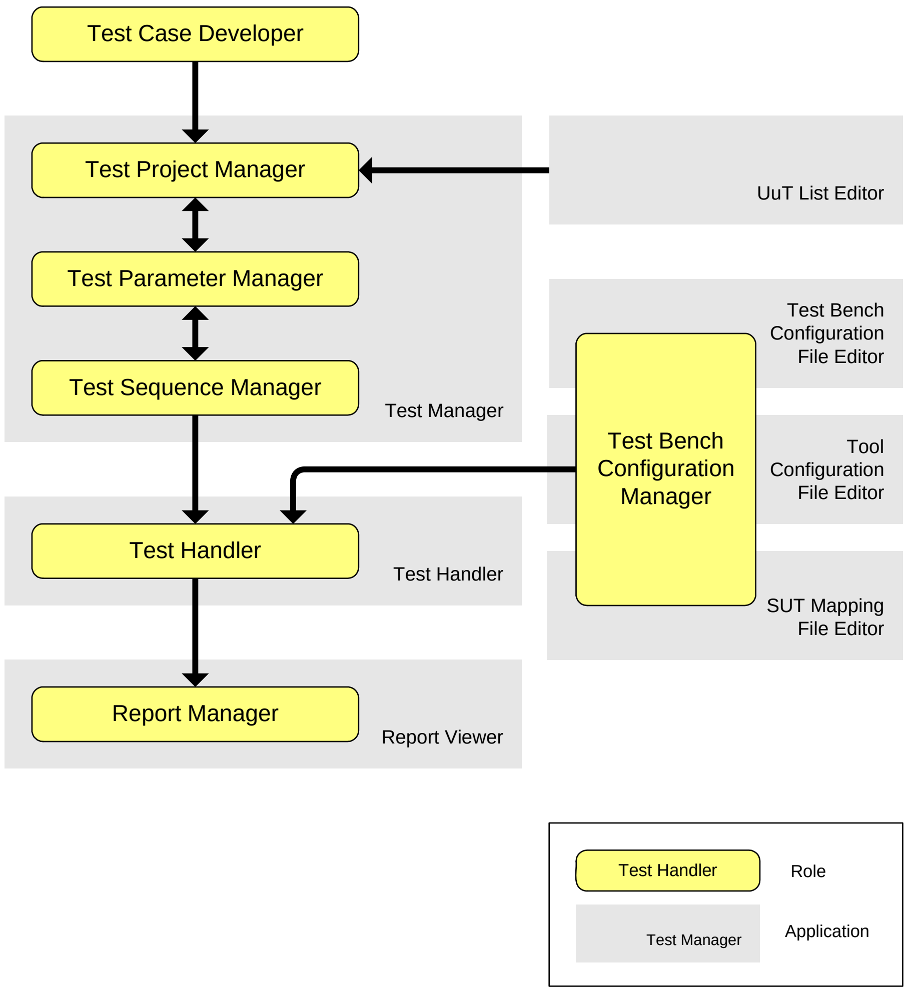


Fig. 1-1 Roles and Applications for their Execution

1.2.1 The Test Case Developer

The Test Case Developer is not part of the task list of LABCAR-AUTOMATION 4.2.3. It develops test-bench-independent test cases and makes these available in a Test Release Library.

Example:

Specification of a vehicle velocity in the DVE model without the following information:

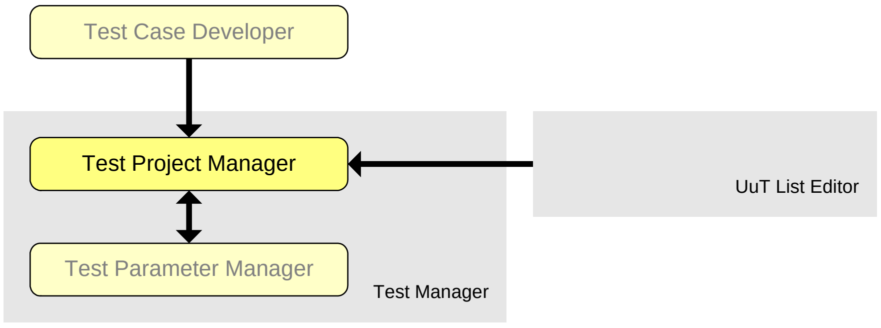
- the location of the velocity in the DVE model
- the physical unit of the vehicle velocity
- the specific value of the velocity which depends on the UuT/project
- tools of the test bench

Implementation:

- Access to the model via the abstract port "P_MA"
- Call of the abstract, tool-independent signature "P_MA.SC_MA_ModelValueSetScalar_SD(label, value, unit)"
- Selection of any physical unit for the velocity
- The parameterization is created later by the Test Parameter Manager, but default, minimum and maximum values are already defined
- The mapping of the abstract label to the physical label takes place later in the test bench configuration

1.2.2 The Test Project Manager

The Test Project Manager (TPrM) is one of the three core roles in LABCAR-AUTOMATION 4.2.3. It creates test projects and manages the UuTs assigned to them.



This role includes the following tasks:

- definition of the logical test project:
  - definition of the structure of the test project
  - definition of the test functionality (i.e. all possible test cases) of the test project.
- assignment of UuTs to the test project
- provision of project-specific information

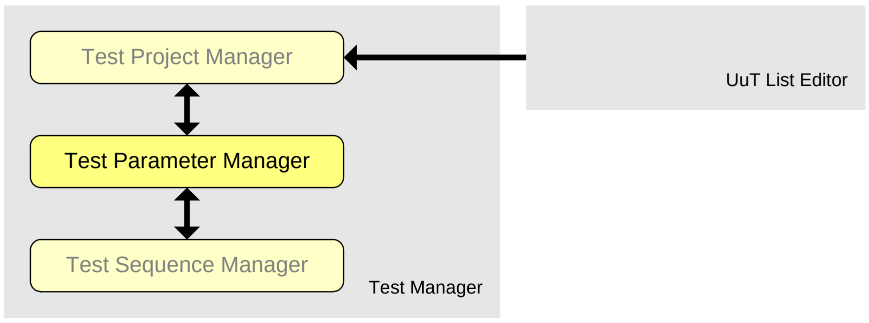
Particularly not part of its tasks are:

- the parameterization of a test case
- topics to do with test execution
- definition of information on the test bench

The workspace of the Test Project Manager is the Test Manager application (see the section "Test Manager" on page 19).

1.2.3 The Test Parameter Manager

Once the Test Project Manager has created the logical test project, it is the task of the Test Parameter Manager (TPM) to transform it into a physical test project.



The tasks of the TPM are particularly:

- the assignment of values to the parameters of a test project
- if necessary, the definition of global parameters
- assignment of UuTs of a test project to individual parameterizations

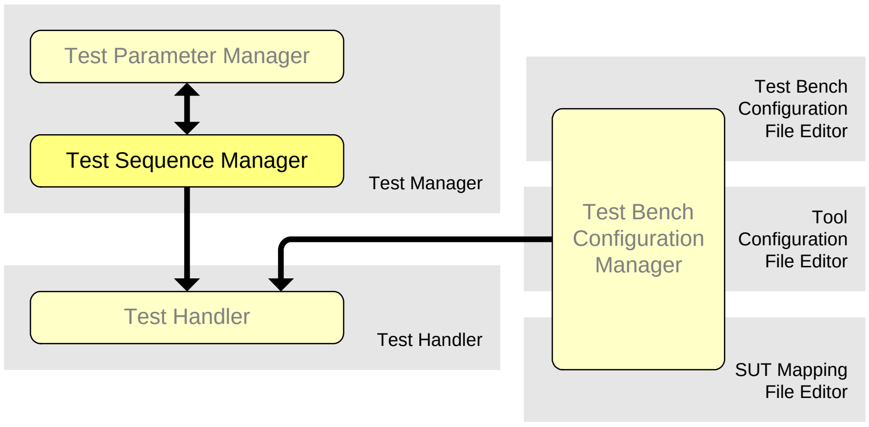
Not part of its tasks are:

- changes to the functionality of the test project
- definition of an execution sequence

The workspace of the Test Parameter Manager is the Test Manager application (see the section "Test Manager" on page 19).

1.2.4 The Test Sequence Manager

After the Test Parameter Manager has created the physical test project, it is the task of the Test Sequence Manager (TSM) to specify execution orders for test runs.



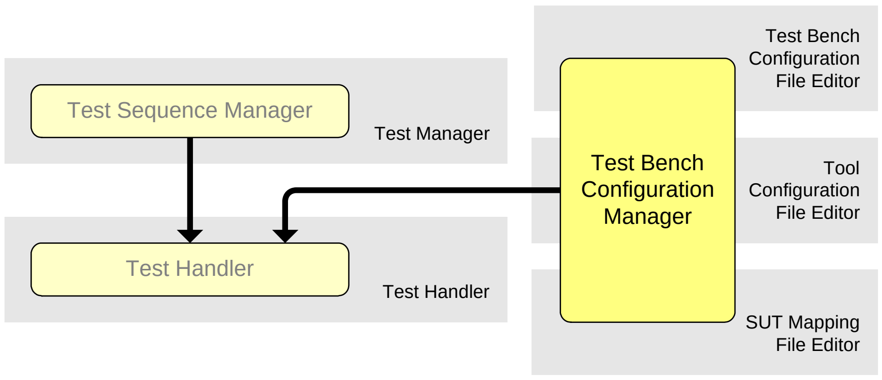
Not part of its tasks are:

- changes to the functionality of the test project
- the parameterization of the test project

The workspace of the Test Sequence Manager is the Test Manager application (see the section "Test Manager" on page 19).

1.2.5 The Test Bench Configuration Manager (TBCM)

The Test Bench Configuration Manager (TBCM) configures all tools required to execute a specific test project.


Test Sequence Manager
Test Manager
Test Handler
Test Handler
Test Bench Configuration Manager
Test Bench Configuration File Editor
Tool Configuration File Editor
SUT Mapping File Editor

This task particularly consists of the provision of the following files:

- Test Bench Configuration File

  The main purpose of the Test Bench Configuration File (TBCF) is to map port instances of the test cases to specific tool instances.

  A description of this file is contained in the section "The Test Bench Configuration File" on page 159.

- Tool Configuration File

  A Tool Configuration File (TCF) contains valid configurations for an application of the test bench (e.g. INCA) required to run the test project.

  A description of this file is contained in the section "The Tool Configuration File" on page 167.

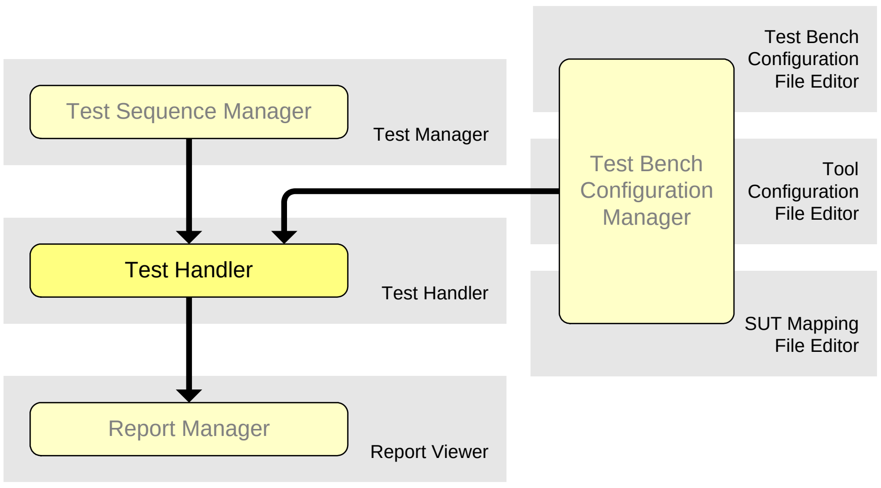
- SUT Mapping File

  This file contains the mapping of the test-bench-independent logical test labels to the labels which are used in the current test environment.

1.2.6 The Test Handler

Once functionality, parameterization and order of execution are defined, the Test Handler (TH) executes the test.

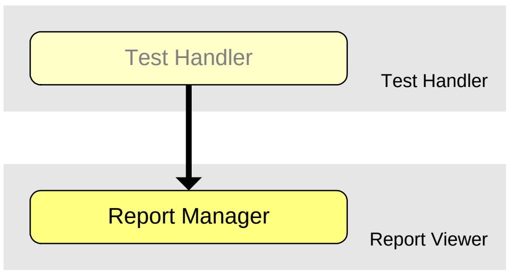
Its most important task is the integration of the test bench the test is to run on.



The workspace of the Test Handler is the Test Handler application (see the section "Test Handler" on page 20).

1.2.7 The Report Manager

The Report Manager determines the display of test reports and manages them.



The workspace of the Report Manager is the Report Viewer application (see the section "Report Viewer" on page 20).

1.3 LABCAR-AUTOMATION 4.2.3 Software Applications

The applications combined in LABCAR-AUTOMATION 4.2.3 have been developed to provide the roles defined in the previous section with an optimum environment and user interface for their relevant tasks within the test project.

When the roles are examined more closely, you can see that particularly three of these roles really do work on the document "test project" as a core task in the sense that they open this project, make corresponding changes to this document in accordance with their role and then resave this project to make it available for further processing.

These three directly project-oriented roles are (see Fig. 1-1 on page 13):

- Test Project Manager (TPrM)

  The TPrM is the product mainly responsible for the test project which it creates and to which it assigns the relevant UuTs.
- Test Parameter Manager (TPM)

  The TPM converts the logical project into a physical project by assigning it a parameterization.
- Test Sequence Manager (TSM)

  The TSM generates executable test sequences from all the available physical tests.

Due to the fact that these three roles work on the same document, there is only one application for these three: Test Manager. All other roles work more or less

- as suppliers for one or more of the core roles described above, such as the Test Case Developer (TCD) or the Test Bench Configuration Manager (TBCM)

or

- as downstream roles in the test process like the Test Handler (TH) or the Report Manager (RM).

Overall, the process with the defined roles is distributed to the five applications shown in Fig. 1-2 which are described in the following sections.

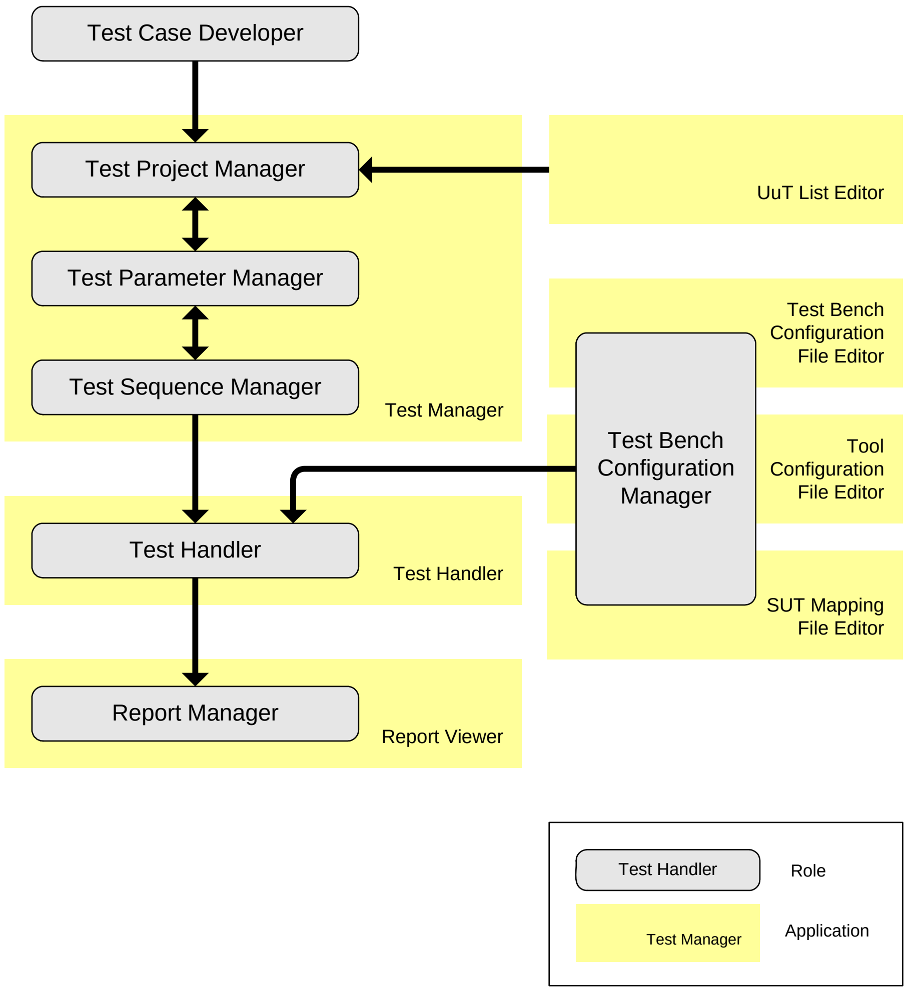


Fig. 1-2 Applications and Roles Working on Them

Test Manager

Test Manager is the application for creating and managing tests, parameterizing them and creating test sequences.

It is the central application on which

- Test Project Manager (see "The Tasks of the Test Project Manager (TPrM)" on page 52),
- Test Parameter Manager (see "The Tasks of the Test Parameter Manager" on page 68) and
- Test Sequence Manager (see "The Tasks of the Test Sequence Manager" on page 92)

work.

Test Handler

Test Handler is the application for the Test Handler. The Test Handler executes defined test sequences on a specific test bench.

A detailed description of the tasks which the Test Handler executes in Test Handler, can be found in the section "The Tasks of the Test Handler" on page 107.

Report Viewer

Report Viewer is the application for defining the type and scope of the reports for the test executed.

A detailed description of the tasks which are processed with Report Viewer, can be found in the section "The Tasks of the Report Manager" on page 134.

1.4 The LABCAR-AUTOMATION 4.2.3 Test Project

The LABCAR-AUTOMATION 4.2.3 test project is the central data container with which the LABCAR-AUTOMATION 4.2.3 test process tools work. Fig. 1-3 shows the structure of this kind of test project.

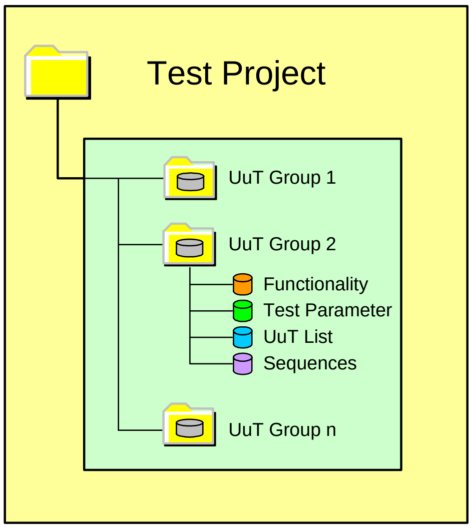


Fig. 1-3 Structure of a LABCAR-AUTOMATION 4.2.3 Test Project

A LABCAR-AUTOMATION 4.2.3 test project consists of any number of what are referred to as "UuT Groups" (see Fig. 1-3).

1.4.1 UuT Groups

In addition to the list of UuTs to be tested ("UuT List"), a UuT Group contains the following objects and data:

- the necessary test cases ("Functionality")
- the relevant test parameters ("Test Parameter" or "Behavior Data")
- the ordered sequences of execution ("Sequences")

1.4.2 Derivates

A derivate is a special parameterization for a UuT within a specific UuT Group.

As already described above, a UuT Group consists of a range of UuTs to be tested for which both test functionality and parameterization have been defined (what is referred to as the Master Data Set of the UuT Group).

In practice it can often happen that specified fixed test functionality has to be separated from test parameterization for certain UuTs. In other words: a lot of UuTs are tested with exactly the same functionality, just with a few deviations in the test parameters of the Master Data Set.

This separation within a UuT Group is made possible by allocating differences in the parameterization to one (or more) individual UuTs. These ”difference parameters” are called derivates.

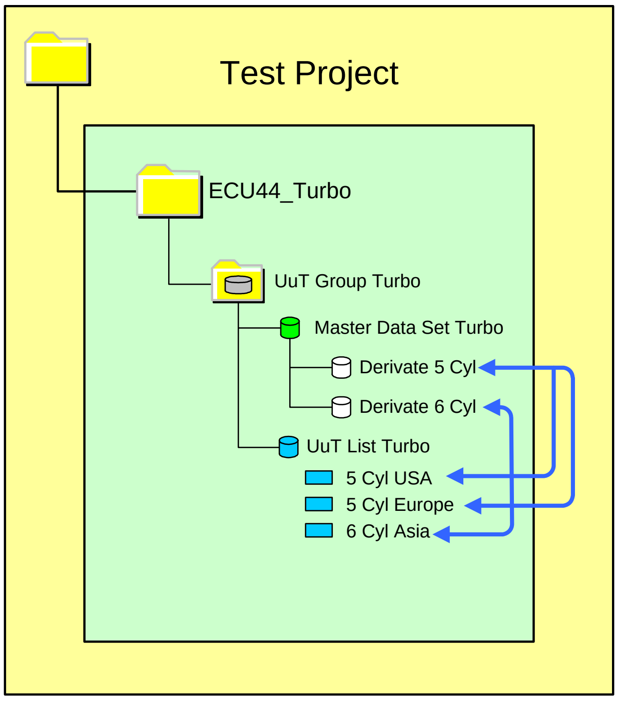


Fig. 1-4 Derivates of a Master Data Set Assigned to Specific UuTs

For more information on working with derivates, turn to the section "Creating Derivates" on page 84.

Overall, a LABCAR-AUTOMATION 4.2.3 test project thus consists of the following components:

- A collection of UuT Groups which contain the objects to be examined (UuTs)
- The definition of test functionality, parameterizations and test sequences for these UuT Groups
- Possibly deviations in the parameterization of individual UuTs within a UuT Group (Derivates)

1.5 Configuration Files

To obtain maximum independence from the test bench tools used when assigning test cases to UuTs, LABCAR-AUTOMATION uses special configuration files which enable threefold ”abstraction”.

1. Abstraction from the Test Bench
2. Abstraction from the Tool
3. Abstraction from Signals and Data

This section contains information on the function of LABCAR-AUTOMATION configuration files.

1.5.1 Abstraction from the Test Bench

The abstraction of the test cases from the test bench used is of top priority here. In the test case, tool-specific commands such as ”Open this LABCAR-OPERATOR experiment”

```
LCO_V3.openExperiment("c:\test\myHilProject.lca")
```

or ”Measure this value in the ECU with INCA”

```
internalECUValue = INCA_V5.getValue("n_eng_Int:devCCP")
```

are not used, but instead ”Open the experiment with the tool for model access”

```
Model_Access.openExperiment(…)
```

or ”Get a value with the tool for ECU access”

```
internalECUValue = ECU_Access.getValue(…)
```

These abstractions are possible for the entire functionality of an HIL test bench, for example for model access, ECU access, etc. These functional groups are referred to as ”ports”.

In addition, a range of operations is defined for every port. These are used instead of direct tool API calls by the test case developer and are called ”signatures”, e.g. ”start()”, ”stop()”, ”setValue(...)” or ”getValue(...)”

This abstraction is resolved by mapping defined ports to specific tools. This information is part of a file - the ”Test Bench Configuration”. The Test Bench Configuration also contains a reference to the ”Tool Configuration”.

For more details on ”Test Bench Configuration”, refer to the section "The Test Bench Configuration File Editor" on page 148.

1.5.2 Abstraction from the Tool

This Tool Configuration is necessary to enable the second level of abstraction: the abstraction of tool-specific information such as special project files and other proprietary configurations.

The Tool Configuration File configures the tool and assigns the abstract model (”Model_ID: IdleControllerMIL”) a specific LABCAR-OPERATOR project (here: ”IdleController.lca”).

In turn, this Model_ID is part of the ”environment” of the UuT - this completes the assignment of test case to UuT. Only the third abstraction (concerning signals and data) has to be resolved.

For more details on ”Tool Configuration”, refer to the section "The Tool Configuration File Editor" on page 163.

1.5.3 Abstraction from Signals and Data

The third phase of abstraction concerns the transfer of data and signals between the test bench and the relevant test case.

Usually, measure values are read from the test system, parameters set or signals recorded in files. All these values are identified by proprietary labels which the test developer cannot know and which can also change over the course of time.

This is why the Test Case Developer does not address this data and these signals directly, but uses abstract test labels which are mapped to the proprietary tool labels in the SUT Mapping File.

For example, the label ”EngineSpeed” is used in the test case:

```
internalECUValue = ECU_Access.getValue("EngineSpeed")
```

While the test is running, this is resolved to a tool-specific label:

```
"EngineSpeed" = "n_eng_Int:devCCP"
```

If labels change in the tool, this is the only place where these changes have to be declared – the test case remains unchanged.

This mapping works for all assignments of the type ”String → String” such as path names, internal options, etc; additional conversions have to be taken into consideration when exchanging numerical data. This concerns, for example, units of measurement such as ”mph” and ”km/h” as well as various temperature scales. This is why a complete SUT Mapping File also has to contain data types, conversion rules, value ranges, etc.

For more details on ”SUT Mapping”, refer to the online help of the SMFE.

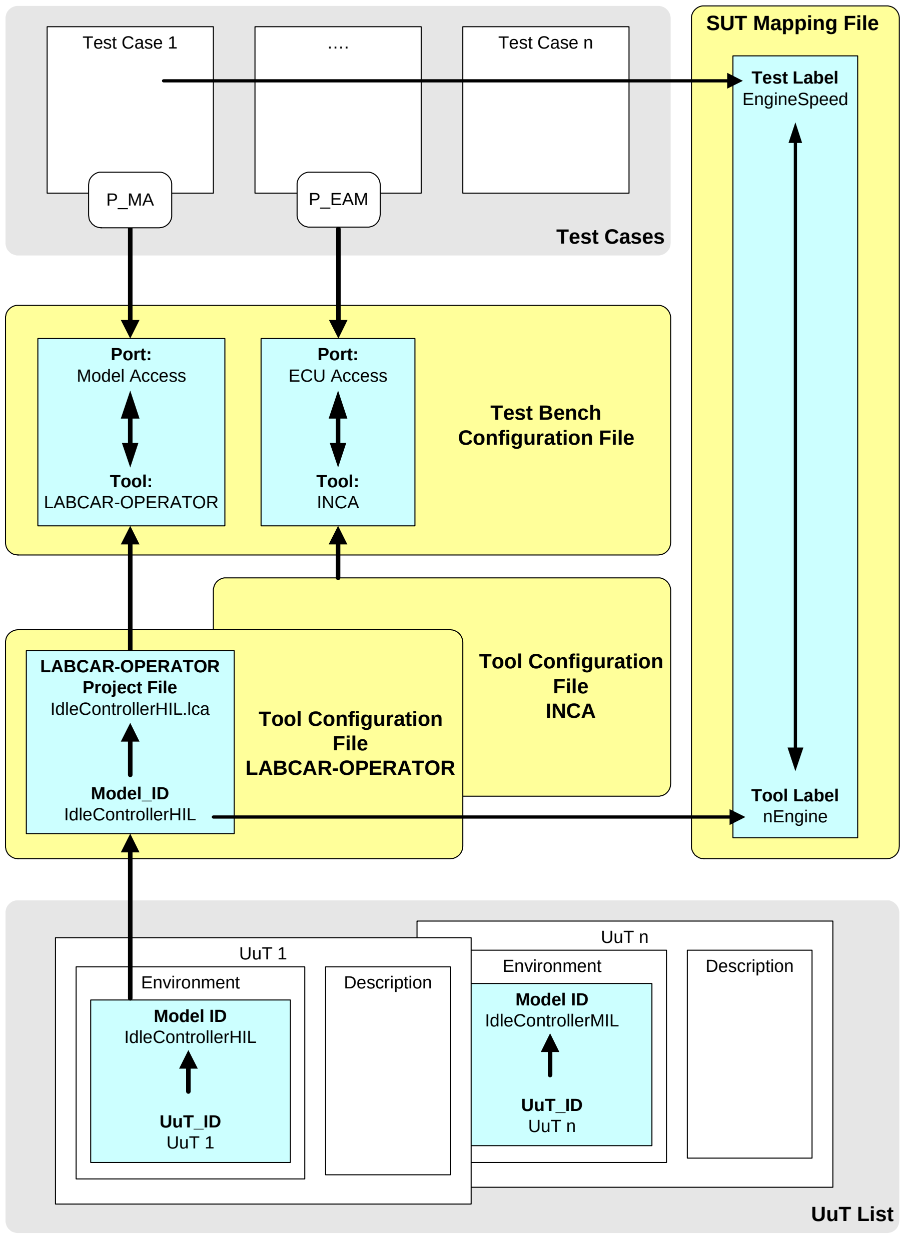


Fig. 1-5 The Configuration Files of LABCAR-AUTOMATION (see Text)

1.5.4 States, Transitions and Configuration Files

Fig. 1-6 shows the state machine using the example of the "P_MA" port for accessing a DVE model that runs in LABCAR-OPERATOR.

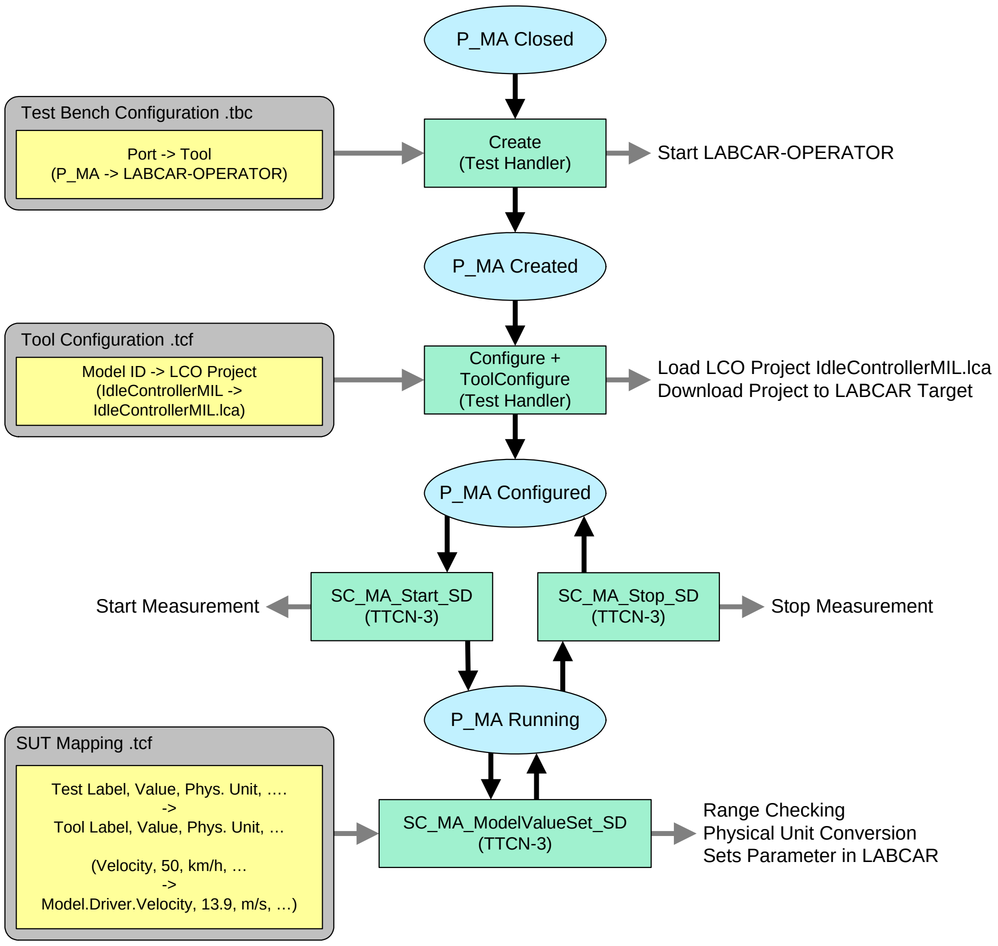


Fig. 1-6 States, Actions and Configuration Files

In the transition from one state to the next, a range of information is required which can be stored in the following places:

- In the test case itself
- In the LABCAR-AUTOMATION project
- In the LABCAR-OPERATOR project (actually: in the "Tool" project)
- In configuration files specified in LABCAR-AUTOMATION:
  - Test Bench Configuration File (TBCF)
  - Tool Configuration File (TCF)
  - SUT Mapping File (SMF)

"P_MA Closed" → "P_MA Created"

This transition is initiated by the Test Handler application. The tool for model access is launched (here: LABCAR-OPERATOR). The basis for this is the definition of the tool for model access in the Test Bench Configuration File. The result is that LABCAR-OPERATOR is launched.

"P_MA Created" → "P_MA Configured"

This transition is also triggered by the Test Handler application. The LABCAR-OPERATOR project is opened and the model downloaded to the target. Information such as the LABCAR-OPERATOR project to be used and the model is required.

In the LABCAR-AUTOMATION project, the abstract model "IdleControllerMIL" is specified for each UuT via the "Model ID". The assignment of the specific LAB-CAR-OPERATOR project "IdleController.lca" (and thus the DVE model) takes place in the Tool Configuration File.

The result is that the LABCAR-OPERATOR project is opened and the model downloaded to the target. This means the "P_MA" port can be accessed in TTCN-3 and, for example, parameters set.

"P_MA Configured" → "P_MA Running"

This transition is triggered by the test case by calling the TTCN-3 signature "SC_MA_StartSD" of the "P_MA" port. This starts and runs simulation (and stops it again when the TTCN-3 signature "SC_MA_StopSD" is called). The information required is, particularly, the mapping of the test labels in the test cases to model labels. This mapping is stored in the SUT Mapping File.

Measuring takes place in the simulation experiment as a result of this transition.

2 Installation

This chapter contains information on the scope of delivery and the system requirements for installation. The installation of any individual product components of LABCAR-AUTOMATION 4.2.3 you have purchased as well as the uninstallation procedure are also described.

You will find information on the following subjects:

- "Preparation" on page 27

  This section contains information on the scope of delivery and the system requirements. Check that the delivery package you receive contains everything it should and that your PC meets the system requirements.
- "Running the Installation" on page 30

  This section describes how to install LABCAR-AUTOMATION 4.2.3.
- "Licensing" on page 34

  To be able to work with LABCAR-AUTOMATION 4.2.3, you require a license file.
- "What is Installed?" on page 37

  This section contains information on the data directories created during installation and their content. It refers to a Windows 7 system environment. When using Windows XP the directories will differ as mentioned below.

2.1 Preparation

This section contains information on the scope of delivery and the system requirements. Check that the delivery package you receive contains everything it should and that your PC meets the system requirements.

2.1.1 Scope of Delivery

LABCAR-AUTOMATION 4.2.3 is supplied with a CD-ROM which contains the following products:

- LABCAR-AUTOMATION 4.2.3 with the following add-ons (to be activated using the relevant software license):
  - Test Automation Core (LCS_LCA_CORE)
  - Test Execution User Interface (LCS_LCA_TE)
  - Test Automation Sequence Builder (LCS_LCA_ASB)
  - Test Manager (LCS_LCA_TM)
  - Test Automation Configuration Wizard Professional (LCS_LCA_CWP)
  - Test Automation Configuration Wizard Standard (LCS_LCA_CWS)
  - Test Automation Editor Package (LCS_LCA_EDP)
  - Test Automation Project Generator (LCS_LCA_PJG)
  - Test Bench Connector for Realtime Testing with Automation (LCS_LCA_TBCRT)
  - Test Bench Connector for ODX Link (LCS_LCA_TBCODX)
  - Test Bench Connector for INTECRIO (LCS_LCA_TBCIRP)

- – Test Bench Connector for Failure Simulation (LCS_LCA_TBCFS)
- – Test Bench Connector for MLBA4 (LCS_LCA_MLBA4)
- – Test Bench Connector for OCT (LCS_LCA_TBCOCT)
- – Test Bench Connector for MDF merger (LCS_LCA_TBMDFM)
- – Test Bench Connector for dSPACE (LCS_LCA_TBCD)

Note

The add-ons are available in packages of varying scope (see "Packages and Licenses" on page 34).

2.1.2 System Requirements

Hardware Requirements

The hardware on which you want to operate LABCAR-AUTOMATION 4.2.3 must fulfill the following requirements:

- 2 GHz processor or higher
- Windows 7, Windows 8.1

  The following language versions of the operating systems listed above are supported:

  - English
  - German
- 4GB RAM; 8 GB RAM recommended
- Hard disk with at least 1 GB of free space

  Note

  Please note when operating LABCAR-AUTOMATION 4.2.3 that report files require a lot of hard-disk memory.
- CD-ROM drive (for the installation)
- VGA graphics card with VGA screen and a resolution of at least 1024 x 768 with 65536 colors

Software Requirements

Depending on the particular application, you will require the following software

- LABCAR-OPERATOR V5.x and possibly add-on products

  For this:
- The correct version of LABCAR-RTPC must be installed on the real-time PC
- Depending on the version of LABCAR-OPERATOR: INCA V6.2 or INCA V7.x.
- ODX-LINK V1.4

Microsoft .NET Framework

Microsoft .NET Framework 2.0 is also required for using LABCAR-AUTOMATION 4.2.3. A check is made during installation to see whether these files are installed. You may be prompted to install Microsoft .NET Framework 2.0.

The installation file is in the start menu of the installation CD under **Tools and Utilities**.

2.1.3 Documentation

Access the User's Guide and various other documents (release notes, online help files etc.) on LABCAR-AUTOMATION 4.2.3 via the "Documentation" link on the start screen of the installation CD.

2.2 Running the Installation

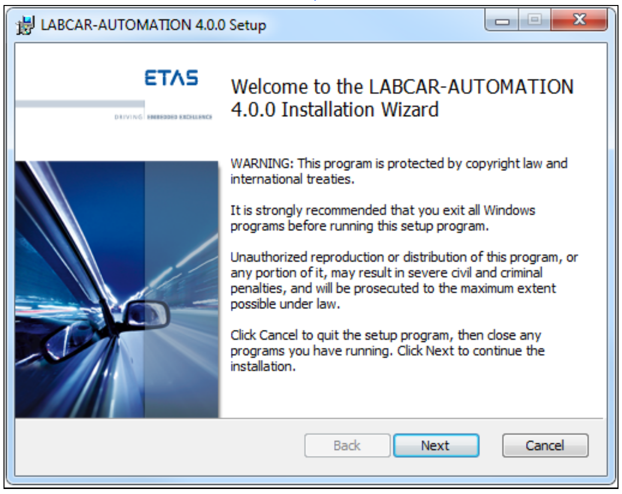
This section describes how to install LABCAR-AUTOMATION 4.2.3.

Note

Uninstall any previous versions of LABCAR-AUTOMATION before you install the current version!

To start installation

- Insert the product CD in your CD-ROM drive.
- From the start screen, select **Installation**.
- The welcome window opens.



Welcome to the LABCAR-AUTOMATION 4.0.0 Installation Wizard

WARNING: This program is protected by copyright law and international treaties.

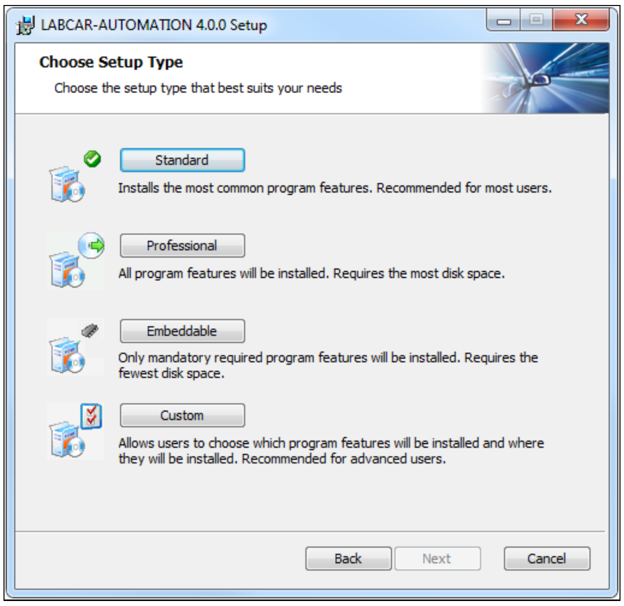
It is strongly recommended that you exit all Windows programs before running this setup program.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Click Cancel to quit the setup program, then close any programs you have running. Click Next to continue the installation.

- Click **Next**.
- Accept the End-User License Agreement in the following window.
- Heed the Safety Hints.

- Select the scope of installation (see "Packages and Licenses" on page 34).


LABCAR-AUTOMATION 4.0.0 Setup

Choose Setup Type

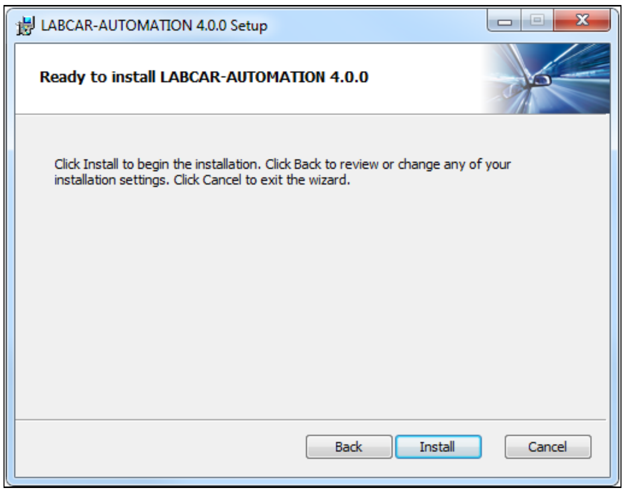
Choose the setup type that best suits your needs

Standard
Installs the most common program features. Recommended for most users.

Professional
All program features will be installed. Requires the most disk space.

Embeddable
Only mandatory required program features will be installed. Requires the fewest disk space.

Custom
Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

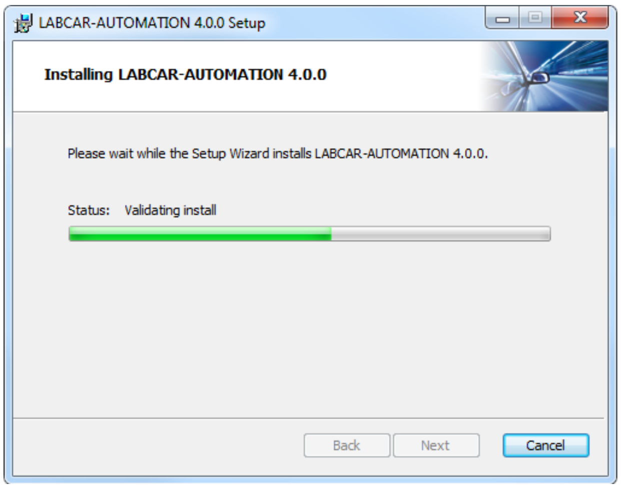

LABCAR-AUTOMATION 4.0.0 Setup

Ready to install LABCAR-AUTOMATION 4.0.0

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

- Click **Install**.

  Installation is executed.



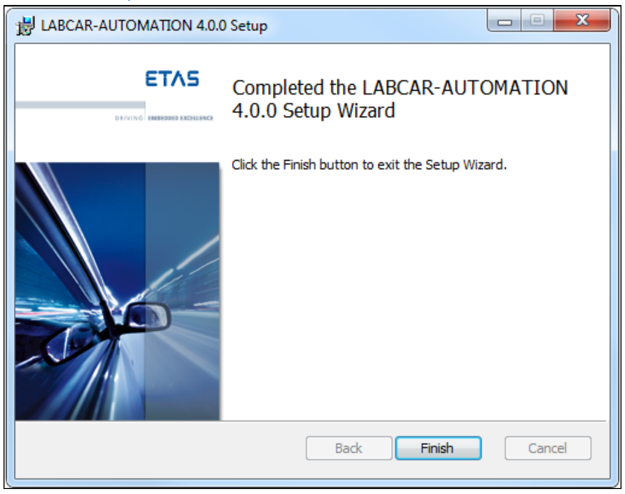
LABCAR-AUTOMATION 4.0.0 Setup

Installing LABCAR-AUTOMATION 4.0.0

Please wait while the Setup Wizard installs LABCAR-AUTOMATION 4.0.0.

Status: Validating install

- To complete installation, click **Finish**.



LABCAR-AUTOMATION 4.0.0 Setup

Completed the LABCAR-AUTOMATION 4.0.0 Setup Wizard

Click the Finish button to exit the Setup Wizard.

2.2.1 Start Menu Folder "LABCAR-AUTOMATION 4.2"

This folder now contains the following entries (depending on the purchased components):

- **Automation Sequence Builder**

  The Automation Sequence Builder supports the graphic-based development of test cases, generates executable code and provides an offline test bench configuration for the validation of the test case.
- **Configuration Wizard**

  The Configuration Wizard is the perfect way to start developing test cases and creating test projects and test bench configurations.
- **Examples**

  Opens a folder containing various example files and projects
- **Manuals**

  Contains a range of documents (manuals, API reference etc.).
- **Report Viewer**

  Launches the application for the Test Report Manager.
- **Test Handler**

  Launches the application for the Test Handler.
- **Test Manager**

  Launches the application for the Test Project Manager, Test Parameter Manager and Test Sequence Manager.
- **Uninstall**

  Starts the uninstallation routine.
- **Test Design (ATCL)**
  - **LABCAR-AUTOMATION Engine Controller**

    Launches the Engine Controller (e.g. for debugging test cases from within Visual Studio).
  - **TCD Generator**

    Generates the necessary files for test cases created with .NET.

2.2.2 Start Menu Folder LABCAR-SMFEditor

If not already available, the latest version of the SUT Mapping File Editor is installed during the installation of LABCAR-AUTOMATION 4.2.3.

The corresponding Start menu folder contains the following entries:

- **SuT Mapping File Editor**

  Launches the SUT Mapping File Editor.
- **SMFE Online Help**

  Launches the SMFE Online Help.

2.3 Licensing

To be able to work with LABCAR-AUTOMATION 4.2.3, you require a license file.

2.3.1 Packages and Licenses

LABCAR-AUTOMATION is available in preconfigured packages (Standard, Professional, Embeddable).

The following components are installed when using these packages:

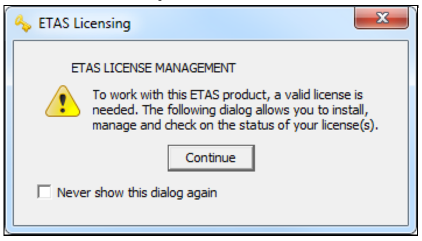
- Standard
  - Test Automation Core (LCS_LCA_CORE)
  - Test Automation Sequence Builder (LCS_LCA_ASB)
  - Test Manager (LCS_LCA_TM)
  - Test Handler (LCS_LCA_TH)
  - Test Automation Configuration Wizard Standard (LCS_LCA_CWS)
  - Test Bench Connector for dSPACE (LCS_LCA_TBCD)
- Professional
  - Test Automation Core (LCS_LCA_CORE)
  - Test Automation Sequence Builder (LCS_LCA_ASB)
  - Test Manager (LCS_LCA_TM)
  - Test Handler (LCS_LCA_TH)
  - Test Automation Editors Package (LCS_LCA_EDP)
  - Test Automation Configuration Wizard Professional (LCS_LCA_CWP)
  - Test Bench Connector for dSPACE (LCS_LCA_TBCD)
  - Test Bench Connector for Realtime Testing with Automation (LCS_LCA_TBCRT)
  - Test Bench Connector for ODX-LINK (LCS_LCA_TBCODX)
  - Test Bench Connector for INTECRIO (LCS_LCA_TBCIRP)
  - Test Bench Connector for Fault Simulation (LCS_LCA_TBCFS)
- Embeddable
  - Test Automation Core (LCS_LCA_CORE)
  - Test Automation Project Generator (LCS_LCA_PJG)
  - Test Bench Connector for OCT (LCS_LCA_TBCOCT)
  - Test Bench Connector for MDF Merger (LCS_LCA_TBMDFM)

The following components can be purchased as an addition to the "Professional" and "Standard" packages and installed using the installation option "Custom":

  - Test Bench Connector for MLBA4 (LCS_LCA_MLBA4)

2.3.2 Managing Licenses with the ETAS License Manager

When you launch a LABCAR-AUTOMATION component for the first time, there is not normally a license installed and the following message is displayed:


ETAS Licensing

ETAS LICENSE MANAGEMENT

To work with this ETAS product, a valid license is needed. The following dialog allows you to install, manage and check on the status of your license(s).

Continue

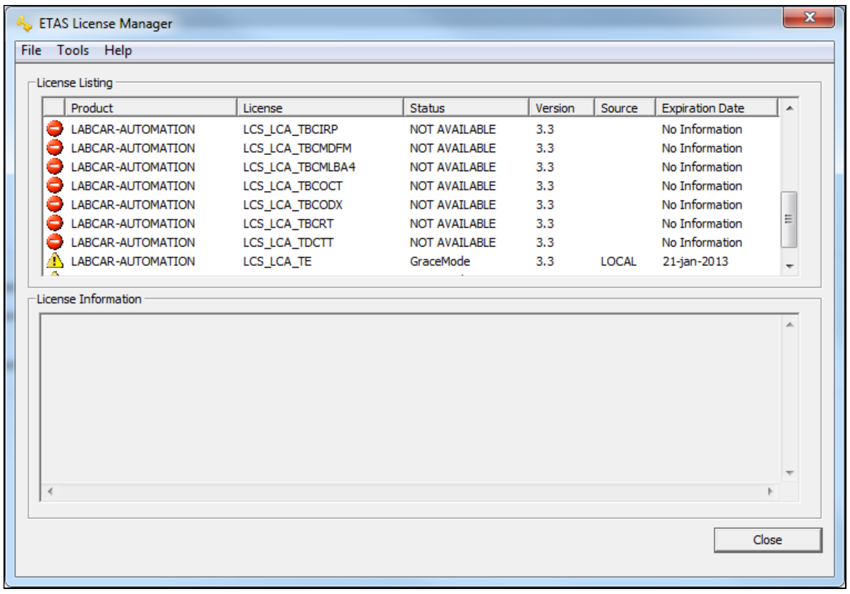
Never show this dialog again

Use the ETAS License Manager to manage licenses: Additional information can be found in the online help of the ETAS License Manager.

To install a license

- Select **Help** → **ETAS License Manager**.

  The ETAS License Manager opens.


ETAS License Manager

File Tools Help

License Listing

	Product	License	Status	Version	Source	Expiration Date
⛔	LABCAR-AUTOMATION	LCS_LCA_TBCIRP	NOT AVAILABLE	3.3		No Information
⛔	LABCAR-AUTOMATION	LCS_LCA_TBCMDFM	NOT AVAILABLE	3.3		No Information
⛔	LABCAR-AUTOMATION	LCS_LCA_TBCMLBA4	NOT AVAILABLE	3.3		No Information
⛔	LABCAR-AUTOMATION	LCS_LCA_TBCOCT	NOT AVAILABLE	3.3		No Information
⛔	LABCAR-AUTOMATION	LCS_LCA_TBCODX	NOT AVAILABLE	3.3		No Information
⛔	LABCAR-AUTOMATION	LCS_LCA_TBCRT	NOT AVAILABLE	3.3		No Information
⛔	LABCAR-AUTOMATION	LCS_LCA_TDCTT	NOT AVAILABLE	3.3		No Information
⚠	LABCAR-AUTOMATION	LCS_LCA_TE	GraceMode	3.3	LOCAL	21-jan-2013

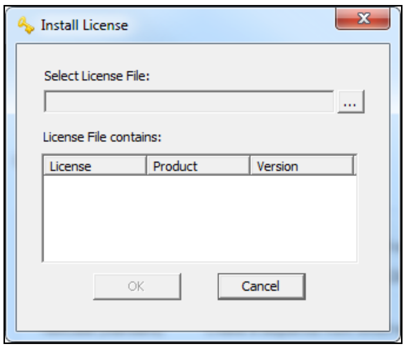
License Information

Close

You can see a list of components which have, as yet, not been licensed.

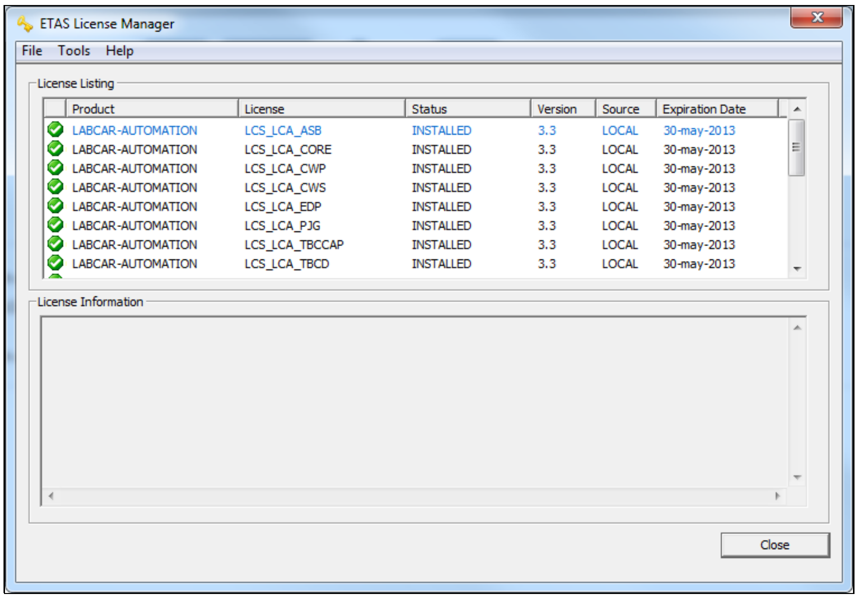
- Select **File** → **Add License File**.

  The “Install License” window opens.



- Click the **...** icon and select the corresponding license file (*.lic). This is usually delivered on the product CD.
- Click **OK** to install the license.

  The licenses are installed.



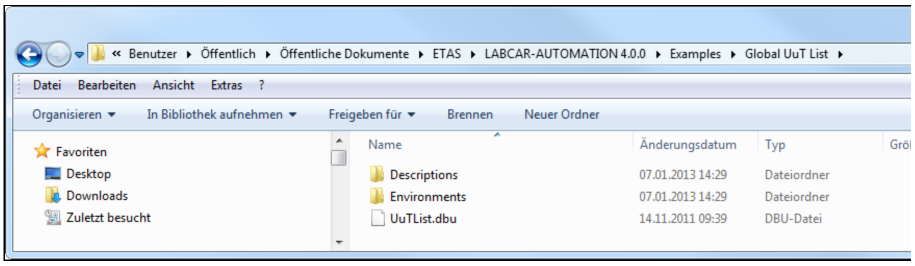
	Product	License	Status	Version	Source	Expiration Date
✓	LABCAR-AUTOMATION	LCS_LCA_ASB	INSTALLED	3.3	LOCAL	30-may-2013
✓	LABCAR-AUTOMATION	LCS_LCA_CORE	INSTALLED	3.3	LOCAL	30-may-2013
✓	LABCAR-AUTOMATION	LCS_LCA_CWP	INSTALLED	3.3	LOCAL	30-may-2013
✓	LABCAR-AUTOMATION	LCS_LCA_CWS	INSTALLED	3.3	LOCAL	30-may-2013
✓	LABCAR-AUTOMATION	LCS_LCA_EDP	INSTALLED	3.3	LOCAL	30-may-2013
✓	LABCAR-AUTOMATION	LCS_LCA_PJG	INSTALLED	3.3	LOCAL	30-may-2013
✓	LABCAR-AUTOMATION	LCS_LCA_TBCCAP	INSTALLED	3.3	LOCAL	30-may-2013
✓	LABCAR-AUTOMATION	LCS_LCA_TBCD	INSTALLED	3.3	LOCAL	30-may-2013

2.4 What is Installed?

This section contains information on the data directories created during installation and their content. It refers to a Windows 7 system environment. When using Windows XP the directories will differ as mentioned below.

2.4.1 The "Users\Public\Documents\ETAS" Folder

During installation, a `\LABCAR-AUTOMATION 4.2` folder is created in this directory which is intended for project-specific data.



Name	Änderungsdatum	Typ	Grö
Descriptions	07.01.2013 14:29	Dateiordner	
Environments	07.01.2013 14:29	Dateiordner	
UuTList.dbu	14.11.2011 09:39	DBU-Datei	

When using Windows XP, this folder will be created in the directory `\Documents and Settings\All Users\Application Data\ETAS`.

The "\Examples" Folder

This folder contains examples of all files used in LABCAR-AUTOMATION projects:

- `\Global UuT List`

  Contains the global UuT list with its descriptions and environments
- `\LCA Test Projects`

  Contains, for example, the tutorial project
- `\LCO Projects`

  Contains various LABCAR-OPERATOR projects
- `\Test Bench Configurations`

  Contains a range of configuration files for LABCAR-AUTOMATION projects
- `\Test Case Development`

  Contains a range of test cases
- `\Test Release Library`

  The Test Release Library (also for the tutorial project)
- `\Test Reports`

  Contains the Test Reports

3 The Test Manager User Interface

Test Manager is the common workspace for the three core roles of LABCAR-AUTOMATION 4.2.3. It must take the main tasks of these core roles into account without providing superfluous information or enabling inadmissible changes.

The main tasks of these three core roles are quickly summarized again below.

Test Project Manager (TPrM)

- The TPrM defines the project structure, i.e. it has to be able to edit it.
- The TPrM defines the scope of the test functionality (test cases used), i.e. it has to be able to edit it.
- The TPrM adds UuTs to UuT Groups, i.e. it has to be able to edit the UuTs.

Test Parameter Manager (TPM)

- The TPM particularly requires a parameter editor to be able to view and to define parameters.
- The TPM must be able to view the project structure to be able to search the project for parameters.
- The TPM must be able to view the test functionality to recognize which context individual parameters are used in.
- The TPM must be able to view the UuT as it defines the assignment between the UuT and the relevant parameterization (derivate).

Test Sequence Manager (TSM)

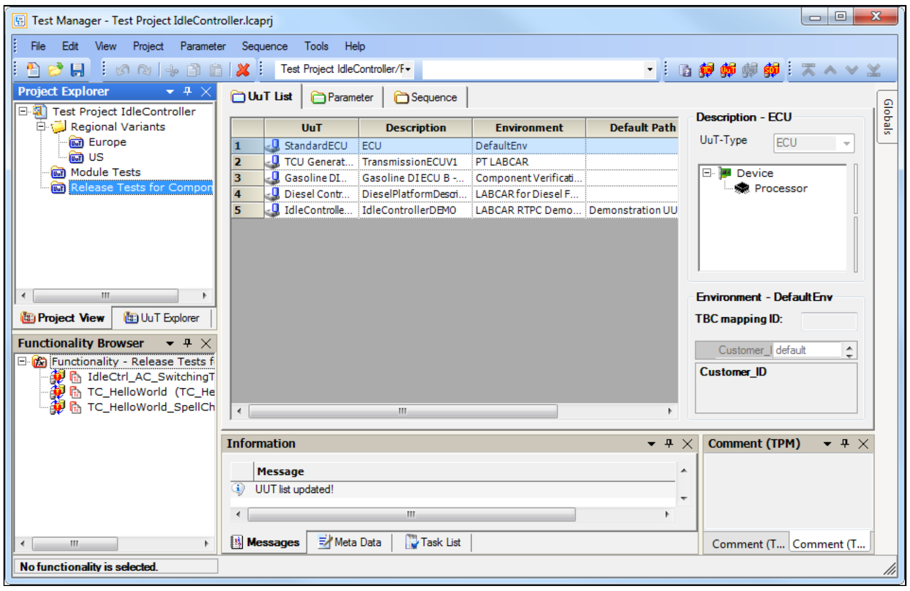
- The TSM particularly requires an editor for defining execution sequences (”sequence editor”).
- The TSM must be able to view the project structure as it defines test sequences for every project level.
- The TSM must be able to view the test functionality as it assigns it within its test sequence.

This chapter contains a general overview of the properties of the user interface – the special tasks of the individual roles within this interface are described in the chapter "Working with LABCAR-AUTOMATION 4.2.3" on page 51.

Fig. 3-1 shows the main window of LABCAR-AUTOMATION 4.2.3. The structure of the interface depends on the requirements described in the previous section:

- The ”Project Explorer” window shows the project structure.
- The Functionality Browser shows the functionality of the object selected in the Project Explorer.
- The role-specific main workspace containing the three tabs ”UuT List”, ”Parameter” and ”Sequence”.
- In the ”Information” window, information on project processing is displayed (”Messages” tab), meta data is shown (”Meta Data” tab) and the Task List (”Task List” tab) is displayed.
- The ”Comment” window contains the tabs ”Comment (TCD)” and ”Comment (TPM)”, in which comments which the relevant roles have given to parameters are displayed and may be edited.

- Global user actions are executed in the "Globals" tab.



	UuT	Description	Environment	Default Path
1	StandardECU	ECU	DefaultEnv	
2	TCU Generat...	TransmissionECUV1	PT LABCAR	
3	Gasoline DI...	Gasoline DI ECU B -...	Component Verificati...	
4	Diesel Contr...	DieselPlatformDescri...	LABCAR for Diesel F...	
5	IdleControlle...	IdleControllerDEMO	LABCAR RTPC Demo...	Demonstration UU

Functionality - Release Tests f
IdleCtrl_AC_SwitchingT
TC_HelloWorld (TC_He
TC_HelloWorld_SpellCh

Message
UUT list updated!

Fig. 3-1 The Main Window of LABCAR-AUTOMATION 4.2.3

Basically the appearance of the user interface is independent of the role of the user - all roles can view the entire project. But what does change according to the role is which data can be edited, for example the Test Project Manager can view parameters but cannot change their values. The individual components of the user interface are described in the following sections.

3.1 The Project Explorer

The Project Explorer displays UuT Groups which make up the LABCAR-AUTOMATION 4.2.3 test project. This kind of UuT Group contains a number of UuTs plus all elements necessary for running the test (see "The LABCAR-AUTOMATION 4.2.3 Test Project" on page 20).

The Project Explorer is used as a starting point for the rough representation of a project – selecting a node in the hierarchy shown determines the representation in all other project views.

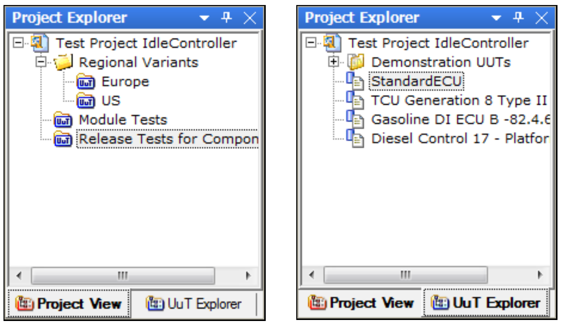


Fig. 3-2 The Project Explorer – "Project View" and "UuT Explorer" Tabs

The Project Explorer has two tabs, "Project View" and "UuT Explorer", which contain two different representations of the project and which thus offer two different possibilities of access to the project:

- "Project View" tab

  The structure of the project is shown with the existing hierarchy levels and the UuT Groups below. The content of the UuT Groups is not shown here.
- "UuT Explorer" tab

  This view shows all UuTs of the project which can be structured in any hierarchy levels by the Test Project Manager.

This allows the user to select three different objects in the Project Explorer:

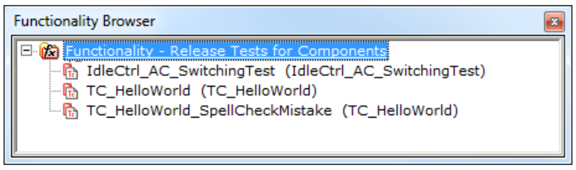
- a hierarchy level of the project
- a UuT Group
- a UuT hierarchy level

3.1.1 Working with the Project Explorer

A detailed description of the functions of the Project Explorer can be found in the section "The Tasks of the Test Project Manager (TPrM)" on page 52.

3.2 The Functionality Browser

This window contains the test functionality of a hierarchy level selected in the Project Explorer, a UuT Group or a UuT.



Functionality - Release Tests for Components
IdleCtrl_AC_SwitchingTest (IdleCtrl_AC_SwitchingTest)
TC_HelloWorld (TC_HelloWorld)
TC_HelloWorld_SpellCheckMistake (TC_HelloWorld)

Fig. 3-3 The Functionality Browser

A detailed description of the functions of the Functionality Browser can be found in the section "The Tasks of the Test Project Manager (TPrM)" on page 52.

3.3 The Main Workspace

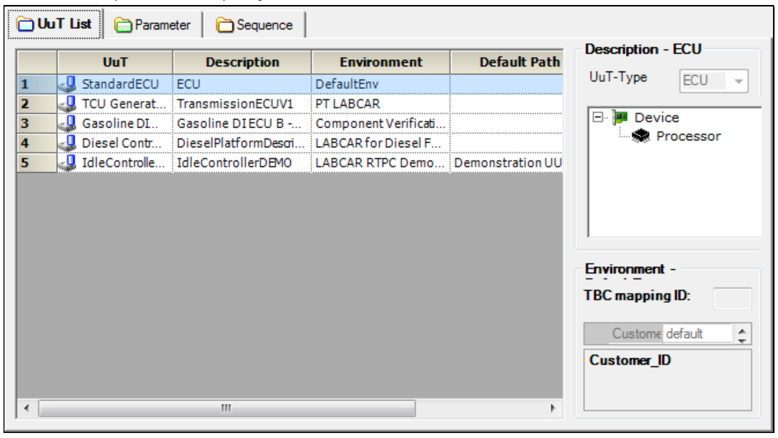
The main workspace has three tabs which contain the workspaces for the three core project roles. This is where the project data is generated, managed and modified according to the role.

The content of the individual tabs depends of course on the elements selected in the Project Explorer and in the Functionality Browser.

Below you will find a short description of the three tabs of the main workspace.

3.3.1 The "UuT List" Tab

This is where the Test Project Manager (TPrM) can add/delete UuTs to/from the UuT Groups of the project.



	UuT	Description	Environment	Default Path
1	StandardECU	ECU	DefaultEnv	
2	TCU Generat...	TransmissionECUV1	PT LABCAR	
3	Gasoline DI...	Gasoline DI ECU B -...	Component Verificati...	
4	Diesel Contr...	DieselPlatformDescri...	LABCAR for Diesel F...	
5	IdleControlle...	IdleControllerDEMO	LABCAR RTPC Demo...	Demonstration UU

Description - ECU

UuT-Type: ECU

Device
Processor

Environment -

TBC mapping ID:

Custome default

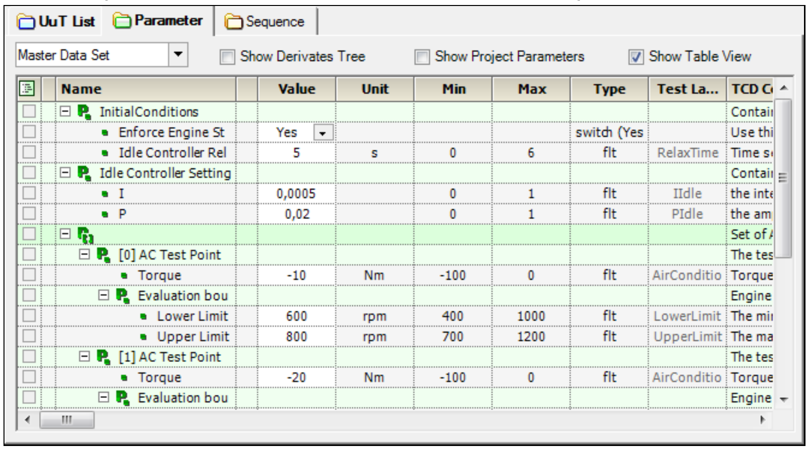
Customer_ID

All "UuT Descriptions" and "UuT Environments" (parameters which are used in the "ConfigureTool" signature of the test development tool) can be viewed in this tab.

A detailed description of this tab is contained in the section "The Tasks of the Test Project Manager (TPrM)" on page 52.

3.3.2 The "Parameter" Tab

This is where the Test Parameter Manager (TPM) edits the parameters of a hierarchical level selected in the Project Explorer (the information of which is also filtered by the relevant selection in the Functionality Browser).



Master Data Set

Show Derivates Tree

Show Project Parameters

Show Table View ✓

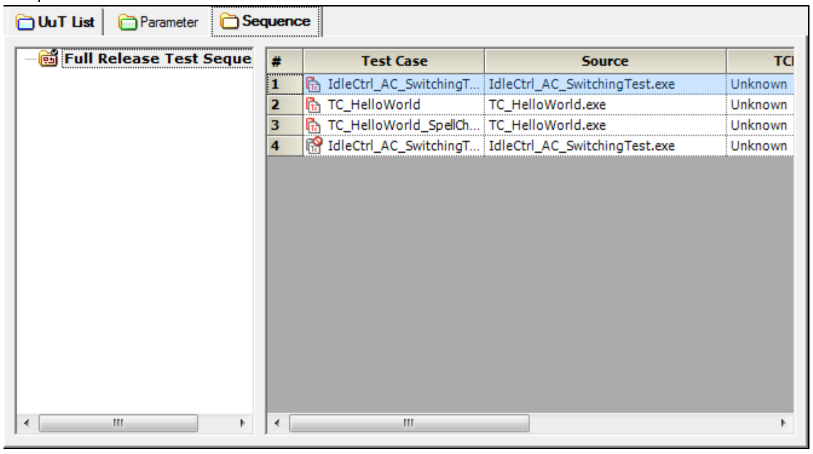
Name	Value	Unit	Min	Max	Type	Test La...	TCD C
InitialConditions							Contai
Enforce Engine St	Yes				switch (Yes		Use thi
Idle Controller Rel	5	s	0	6	flt	RelaxTime	Time s
Idle Controller Setting							Contai
I	0,0005		0	1	flt	IIdle	the inte
P	0,02		0	1	flt	PIdle	the am
							Set of
[0] AC Test Point							The tes
Torque	-10	Nm	-100	0	flt	AirConditio	Torque
Evaluation bou							Engine
Lower Limit	600	rpm	400	1000	flt	LowerLimit	The mi
Upper Limit	800	rpm	700	1200	flt	UpperLimit	The ma
[1] AC Test Point							The tes
Torque	-20	Nm	-100	0	flt	AirConditio	Torque
Evaluation bou							Engine

The content of the tab consists of a table which is created from the parameters (rows) and their attributes (columns).

A detailed description of this tab is contained in the section "The Tasks of the Test Parameter Manager" on page 68.

3.3.3 The "Sequence" Tab

This is where the Test Sequence Manager defines the test sequences by adding or canceling functionality, changing the order of execution and deactivating test steps.



#	Test Case	Source	TC
1	IdleCtrl_AC_SwitchingT...	IdleCtrl_AC_SwitchingTest.exe	Unknown
2	TC_HelloWorld	TC_HelloWorld.exe	Unknown
3	TC_HelloWorld_SpellCh...	TC_HelloWorld.exe	Unknown
4	IdleCtrl_AC_SwitchingT...	IdleCtrl_AC_SwitchingTest.exe	Unknown

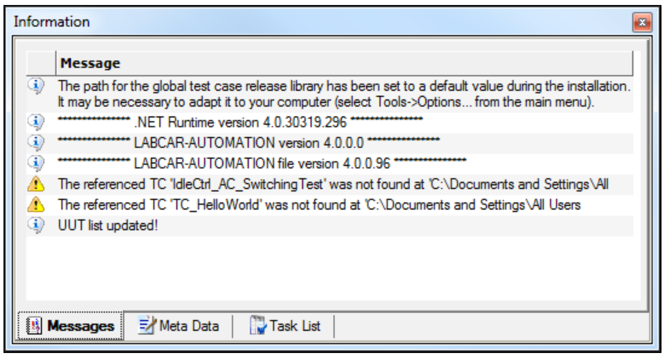
A detailed description of this tab is contained in the section "The Tasks of the Test Sequence Manager" on page 92.

3.4 The "Information" Window

This window has three tabs in which messages on project processing are displayed ("Messages" tab), meta data is shown ("Meta Data" tab) and the Task List ("Task List" tab) is displayed.

3.4.1 The "Messages" Tab

This tab shows the user messages such as warnings, errors and other information. Using the shortcut menu of this tab, the list can be cleared (**Clear Log**), filtered (**Filter** →) or saved (**Save as**).



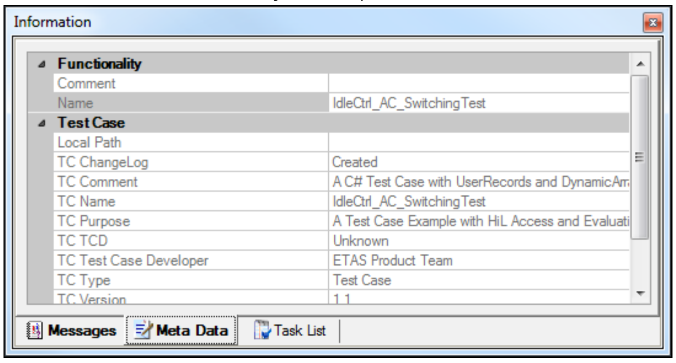
```
Message
The path for the global test case release library has been set to a default value during the installation.
It may be necessary to adapt it to your computer (select Tools->Options... from the main menu).
*************** .NET Runtime version 4.0.30319.296 ***************
*************** LABCAR-AUTOMATION version 4.0.0.0 ***************
*************** LABCAR-AUTOMATION file version 4.0.0.96 ***************
The referenced TC 'IdleCtrl_AC_SwitchingTest' was not found at 'C:\Documents and Settings\All
The referenced TC 'TC_HelloWorld' was not found at 'C:\Documents and Settings\All Users
UUT list updated!
```

Fig. 3-4 The "Messages" Tab

3.4.2 The "Meta Data" Tab

This tab shows object-specific meta data (in Fig. 3-5 meta data of a test case) – for objects which were selected in the Project Explorer and the Functionality Browser.

The meta data can refer to the project selected in the Project Explorer or to a test case selected in the Functionality Browser. The meta data is structured into "sections" which contain key/value pairs.



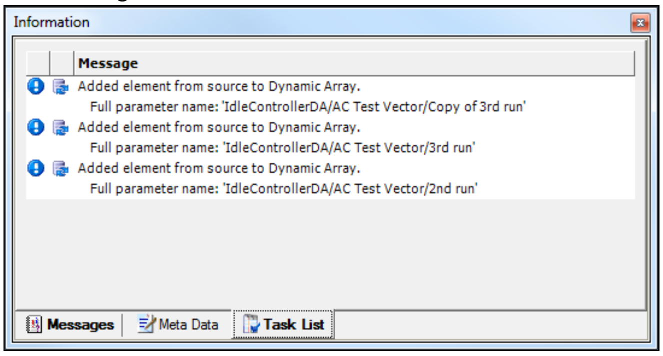
Functionality	
Comment	
Name	IdleCtrl_AC_SwitchingTest
Test Case	
Local Path	
TC ChangeLog	Created
TC Comment	A C# Test Case with UserRecords and DynamicArr
TC Name	IdleCtrl_AC_SwitchingTest
TC Purpose	A Test Case Example with HiL Access and Evaluati
TC TCD	Unknown
TC Test Case Developer	ETAS Product Team
TC Type	Test Case
TC Version	1.1

Fig. 3-5 The "Meta Data" Tab

When a project is created, each object has a predefined meta data structure ("system defined meta data") – this data has to be specified when the relevant object is created (such as "Project Group" with a project) or it is already predefined (as for example "Test Case Purpose" which has already been defined by the Test Case Developer).

3.4.3 The "Task List" Tab

This tab displays error messages which occur when dynamic arrays are inserted from the clipboard and when parameter sets are imported. If a message refers to a parameter in the list of parameters, it is highlighted in the parameter list when the message is double-clicked.



Message
Added element from source to Dynamic Array. Full parameter name: 'IdleControllerDA/AC Test Vector/Copy of 3rd run'
Added element from source to Dynamic Array. Full parameter name: 'IdleControllerDA/AC Test Vector/3rd run'
Added element from source to Dynamic Array. Full parameter name: 'IdleControllerDA/AC Test Vector/2nd run'

Fig. 3-6 The "Task List" Tab

3.5 The "Comments" Window

This window contains three tabs in which comments of specific roles on a selected parameter are displayed.

- Comments (TCD)

  Comments from the Test Case Developer are displayed here. These cannot be edited by any role in LABCAR-AUTOMATION.

- Comments (TPM)

  The Test Parameter Manager can enter comments in this tab.

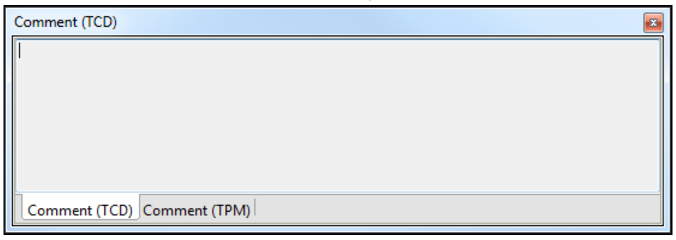


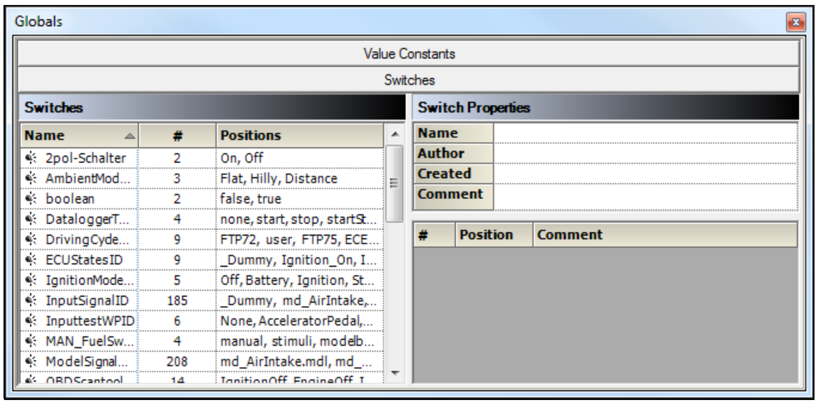
Fig. 3-7 The "Comment" Window

3.6 The "Globals" Window

This window is made visible by clicking the "Globals" tab at the top right edge of the Test Manager user interface.

3.6.1 Switches

All the switches contained in the project-specific Switch Definition File and their definition are displayed under "Switches". This file can be selected using **Project → Options** under "File Locations".



Name	#	Positions
2pol-Schalter	2	On, Off
AmbientMod...	3	Flat, Hilly, Distance
boolean	2	false, true
DataloggerT...	4	none, start, stop, startSt...
DrivingCycle...	9	FTP72, user, FTP75, ECE...
ECUStatesID	9	_Dummy, Ignition_On, I...
IgnitionMode...	5	Off, Battery, Ignition, St...
InputSignalID	185	_Dummy, md_AirIntake,...
InputtestWPID	6	None, AcceleratorPedal,...
MAN_FuelSw...	4	manual, stimuli, modelb...
ModelSignal...	208	md_AirIntake.mdl, md_...
OBDScantool	14	IgnitionOff, EngineOff, I

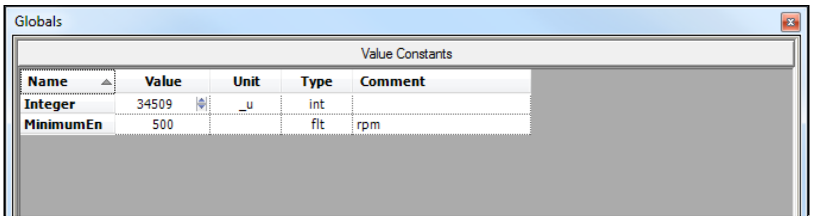
Name	
Author	
Created	
Comment	

#	Position	Comment

Fig. 3-8 The "Globals - Switches" Window

3.6.2 Value Constants

All constant definitions valid for the project created by the Test Parameter Manager (TPM) are displayed under "Value Constants".



Name	Value	Unit	Type	Comment
Integer	34509	_u	int	
MinimumEn	500		flt	rpm

Fig. 3-9 The "Globals - Value Constants" Window

3.7 The Test Manager Main Menu

The main menu consists of eight items:

- **File**
  See "The "File" Menu" on page 46.
- **Edit**
  See "The "Edit" Menu" on page 47.
- **View**
  See "The "View" Menu" on page 47.
- **Project**
  See "The "Project" Menu" on page 47.
- **Parameter**
  See "The "Parameter" Menu" on page 48.
- **Sequence**
  See "The "Sequence" Menu" on page 48.
- **Tools**
  See "The "Tools" Menu" on page 48.
- **Help**
  See "The "Help" Menu" on page 48.

3.7.1 The "File" Menu

- **File → New**
  This is used to create a new project or a new test case collection.
- **File → Open**
  Opens an existing project by selecting the project file.
- **File → Open by Attributes**
  Opens a dialog box in which projects can be searched for using specific attributes.
- **File → Close**
  Closes the open project.
- **File → Save**
  Saves the current project.
- **File → Save as**
  Saves the current project under a different name.
- **File → Recent Projects →**
  Shows a list of the projects last opened. A selected project is opened.
- **File → Exit**
  Exits Test Manager.

3.7.2 The "Edit" Menu

- **Edit** → *Name of the action to be undone*

  Undoes the action last executed.
- **Edit** → *Name of the action to be repeated*

  Returns an undone action to the original state.
- **Edit → Cut**

  Cuts functionality, parameter or sequences (depending on the context) and saves them in the clipboard.
- **Edit → Copy**

  Copies functionality, parameter or sequences (depending on the context) into the clipboard.
- **Edit → Paste**

  Adds functionality, parameter or sequences (depending on the context) from the clipboard.
- **Edit → Rename**

  Makes renaming of an object possible.
- **Edit → Delete**

  Deletes the selected object.

3.7.3 The "View" Menu

- **View → Project Explorer**

  Opens/closes the "Project Explorer" window.
- **View → Functionality Browser**

  Opens/closes the Functionality Browser.
- **View → Globals**

  Opens/closes the "Globals" window.
- **View → Information Window**

  Opens/closes the "Information" window.
- **View → Parameter Comments**

  Opens/closes the "Comments" window.

3.7.4 The "Project" Menu

- **Project → Options**

  Opens a dialog box in which project options can be set.
- **Project → Synchronize →**
  - **Check TRL for updates**

    Checks whether there are more recent versions of the test cases used in the project in the Test Release Library (see "Synchronization" on page 66).

- **Test Cases**
  For synchronizing test cases.
- **UuT List**
  For synchronizing UuT Lists.
- **Switch Definition File**
  For synchronizing Switch Definition Files.
- **Unit Conversion Table**
  For synchronizing Unit Conversion Tables.

3.7.5 The "Parameter" Menu

- **Parameter → Customize Attribute View**
  Opens a dialog box in which the display of attributes can be adapted to user requirements.
- **Parameter → Search**
  Opens a dialog box in which parameters can be searched for using names or parts of names.

3.7.6 The "Sequence" Menu

- **Sequence → Move to Top**
  Positions one or more test sequences selected in the "Sequence" tab at the start of the list.
- **Sequence → Move Up**
  Moves one or more test sequences selected in the "Sequence" tab one position up.
- **Sequence → Move Down**
  Moves one or more test sequences selected in the "Sequence" tab one position down.
- **Sequence → Move to Bottom**
  Positions one or more test sequences selected in the "Sequence" tab at the end of the list.

3.7.7 The "Tools" Menu

- **Tools → Options**
  This is where various global settings such as paths and files can be set (independently of the project currently open).

3.7.8 The "Help" Menu

- **Help → User's Guide**
  Opens the English User's Guide (PDF).

- **Help → System Info**

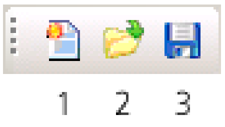
  Opens an information window containing details of the current versions of the components of LABCAR-AUTOMATION 4.2.3.
- **Help → About**

  Opens a window containing general information.

3.8 The Test Manager Toolbars

There are a number of icons, corresponding to frequently used menu items, contained in the toolbars. The icons are used for the following functions:

3.8.1 The "File" Toolbar



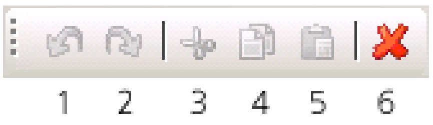
1. **New**

   Exactly the same as **File → New**.
2. **Open**

   Exactly the same as **File → Open**.
3. **Save**

   Exactly the same as **File → Save**.

3.8.2 The "Edit" Toolbar



1. **Undo**

   Exactly the same as **Edit → Undo**.
2. **Redo**

   Exactly the same as **Edit → Redo**.
3. **Cut**

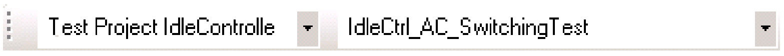
   Exactly the same as **Edit → Cut**.
4. **Copy**

   Exactly the same as **Edit → Copy**.
5. **Paste**

   Exactly the same as **Edit → Paste**.
6. **Delete**

   Exactly the same as **Edit → Delete**.

3.8.3 The "Project" Toolbar



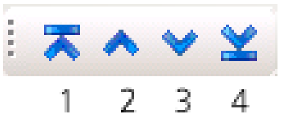
1. **ProjectHierarchies**

   Makes it possible to select individual hierarchy levels in the project.

2. **FunctionalityHierarchies**

   If a UuT Group is selected in the "ProjectHierarchies" selection, the functionality of this UuT Group can be selected here.

3.8.4 The "Sequence" Toolbar



1. **Move to Top**

   Exactly the same as **Sequence** → **Move to Top**.

2. **Move Up**

   Exactly the same as **Sequence** → **Up**.

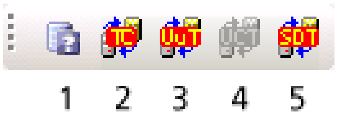
3. **Move Down**

   Exactly the same as **Sequence** → **Down**.

4. **Move to Bottom**

   Exactly the same as **Sequence** → **Move to Bottom**.

3.8.5 The "Synchronization" Toolbar



1. **Check TRL for updates**

   See **Project** → **Synchronize** → **Check TRL for updates**.

2. **Synchronize Test Cases**

   See **Project** → **Synchronize** → **Test Cases**.

3. **Synchronize UuT List**

   See **Project** → **Synchronize** → **UuT List**.

4. **Synchronize Unit Conversion Table**

   See **Project** → **Synchronize** → **Unit Conversion Table**.

5. **Synchronize Switch Definition File**

   See **Project** → **Synchronize** → **Switch Definition File**.

4 Working with LABCAR-AUTOMATION 4.2.3

This chapter contains a description of how the tasks of individual roles are carried out when executing a test project.

It includes information on the following topics:

- "The Tasks of the Test Project Manager (TPrM)" on page 52

  The Test Project Manager creates test projects, assigns UuTs to them and defines the test functionality.

- "The Tasks of the Test Parameter Manager" on page 68

  The Test Parameter Manager (TPM) transfers a logical test project (defined by the Test Project Manager) into a physical test project. This particularly means that it assigns all parameters (defined by the Test Case Developer) suitable values or releases the default values assigned by the Test Case Developer.

- "The Tasks of the Test Sequence Manager" on page 92

  The Test Sequence Manager determines orders of execution and all other details which the Test Handler needs to execute a test for test cases of a UuT Group which have been assigned parameters.

- "The Tasks of the Test Bench Configuration Manager" on page 98

  The Test Bench Configuration Manager (TBCM) provides all tools to make a specific test project executable.

- "The Tasks of the Test Handler" on page 107

  The Test Handler's job is to execute the test. This means that it selects a UuT and thus implicitly a list of test sequences. Together with a valid Test Bench Configuration, it is then capable of carrying out test sequences. Test reports are created automatically once the sequences have been executed.

- "The Tasks of the Report Manager" on page 134

  The Report Manager is responsible for displaying test reports saved in XML format which were generated when a test sequence was executed.

4.1 The Tasks of the Test Project Manager (TPrM)

The Test Project Manager creates test projects, assigns UuTs to them and defines the test functionality.

The main application of LABCAR-AUTOMATION 4.2.3, Test Manager is used for this purpose.

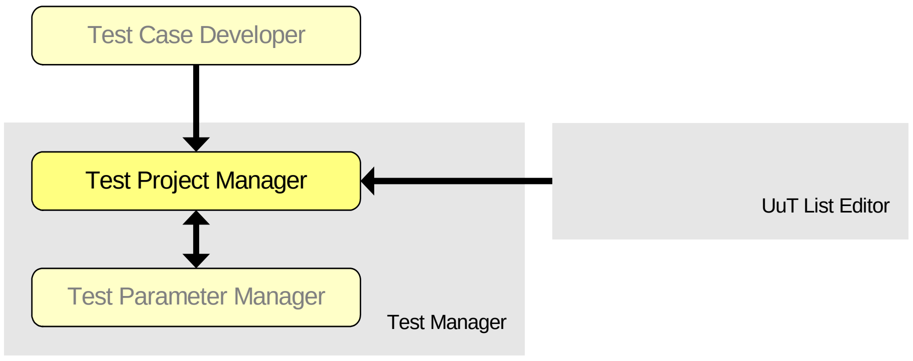


Fig. 4-1 The Test Project Manager

Below, you will find information on executing the different tasks of the TPrM.

The individual sections are:

- "Creating Test Projects" on page 53
- "Working with the Project Explorer - ”Project View” Tab" on page 54
- "Working with the Project Explorer - ”UuT Explorer” Tab" on page 55
- "Managing UuTs" on page 56
- "Adding Test Functionality" on page 60
- "Selection of the Switch Definition File" on page 65
- "Synchronization" on page 66

4.1.1 Creating Test Projects

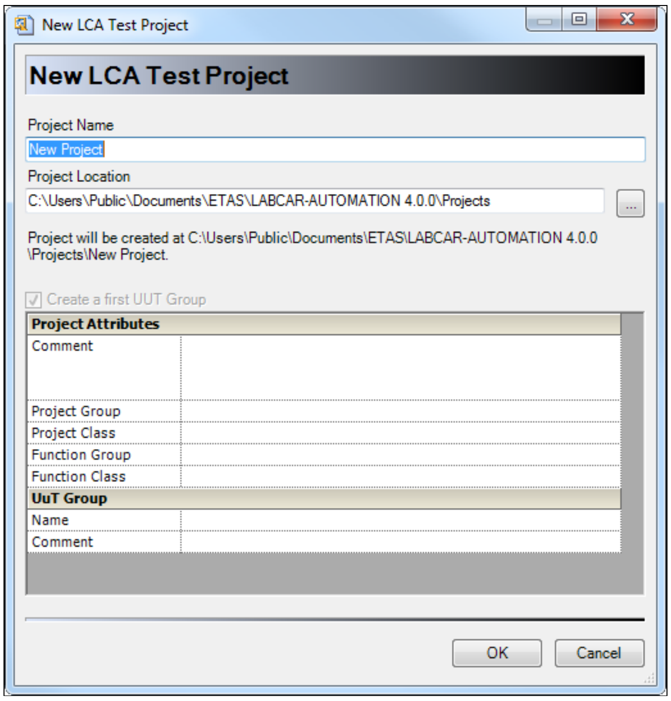
Proceed as follows to create a new test project:

To launch Test Manager

- Select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2** → **Test Manager** from the Windows Start menu.
- Test Manager is launched.

To create a test project

- Close any open projects.
- Select **File** → **New**.
- The "New LCA Test Project" dialog box opens.



- Enter the following information:

Note

The information marked with an asterisk (*) is mandatory!

  - The name of the project* ("Project Name")
  - The directory* in which the new project is to be generated ("Project Location").

    Use the **...** button to open a directory selection window.

- The project attributes listed, "Comment", "Project Group", "Project Class", "Function Group" and "Function Class".
- The name of the UuT Group to be created* ("UuT Group - Name").

Note

Any combination of these four attributes has to be unambiguous within the project so that the project can be identified via these attributes.

- Click **OK**.

  The project files are created and the project is displayed in the Project Explorer.

Working with the Project Explorer - "Project View" Tab

The Project Explorer is the main window of the Test Project Manager – this is where the test project is managed with its hierarchy.

The display in the Project Explorer starts with the project ("ICPR059", see Fig. 4-2); below that there are the UuT Groups (see "To add a UuT to the UuT Group" on page 57) to which, in turn, the individual UuTs are assigned. The latter do not, however, appear in this view but are listed in the "UuT List" tab of the main workspace (see "Managing UuTs" on page 56).

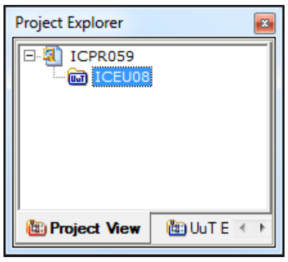
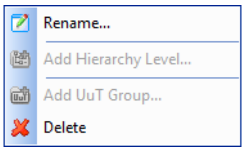


Fig. 4-2 The Project Explorer -"Project View" Tab

The selection of an element in the Project Explorer (such as a UuT Group or a hierarchy level) also influences the content of the other windows of the user interface (apart from "Globals").

The Functionality Browser shows the available test functionality for the selected hierarchy level, the "Information" window shows the meta data of the selected hierarchy level and the main window – depending on the tab selected – displays either UuTs ("UuT List"), parameters ("Parameter") or test sequences ("Sequence") contained.

The project structure is edited via the shortcut menu of the Project Explorer.



The shortcut menu contains the following items:

- **Rename**

  This is used to rename the selected folder or UuT Group.
- **Duplicate**

  Creates a duplicate of the selected folder or the selected UuT Group with all elements (lower in the hierarchy) apart from UuTs.
- **Add Hierarchy Level**

  Creates a new hierarchy level (folder).
- **Add UuT Group**

  Adds a new (empty) UuT Group to the project. If a UuT Group is selected, this menu item is deactivated.
- **Delete**

  Deletes the selected node together with all the elements in it.

Working with the Project Explorer - "UuT Explorer" Tab

The second tab of the Project Explorer, "UuT Explorer", offers a second view of the project in the form of what is referred to as the UuT structure.

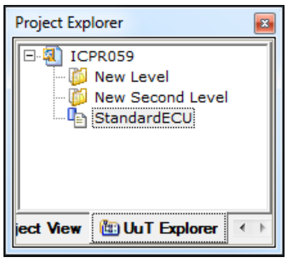


Fig. 4-3 The "UuT Explorer" Tab: UuT Structure and UuTs

This structure can be defined by the TPrM independently of other hierarchy information – it is only used to sort the existing UuTs in accordance with certain criteria. This means the hierarchy structure shown in the "Project View" tab is not visible in this view; only individual UuTs are displayed.

Note

The UuTs themselves cannot be edited or modified here – only the situation of the UuTs within the structure defined previously can be modified.

Selecting a UuT in this tab corresponds to selecting a UuT Group. All objects and properties (such as functionality, parameters, test sequences etc.) belonging to this UuT Group are displayed in the other windows and their tabs.

A UuT can be assigned to a folder (i.e., to certain level within the hierarchy) using drag-and-drop.

There is a shortcut menu for creating and managing the UuT structure; it is shown in Fig. 4-4.

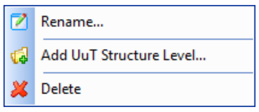


Fig. 4-4 The Shortcut Menu of the "UuT Explorer" Tab

- **Rename**

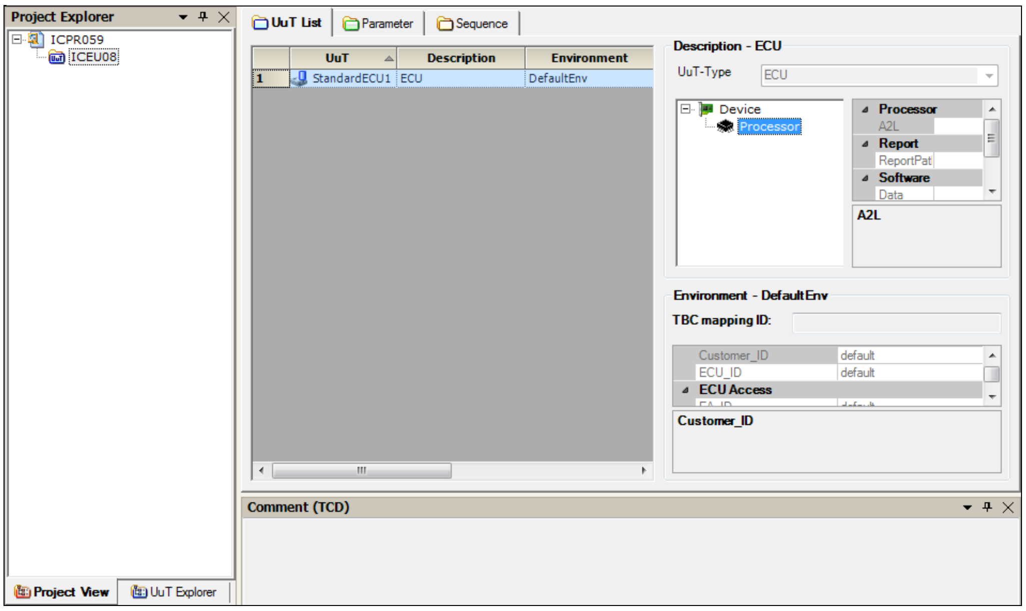
  Renames the selected hierarchy level - UuTs themselves cannot be renamed at this point.
- **Add UuT Structure Level**

  Adds a new hierarchy level at the selected point.
- **Delete**

  Deletes the selected node with all the elements contained within the node.

4.1.2 Managing UuTs

One of the tasks of the Test Project Manager is the assignment of UuTs to the test project. How to deal with UuT Groups has already been described above – the UuTs within the UuT Groups are managed in the main workspace in the "UuT List" tab (see Fig. 4-5).



	UuT	Description	Environment
1	StandardECU1	ECU	DefaultEnv

Description - ECU

UuT-Type: ECU

Device
Processor

Processor	
A2L	
Report	
ReportPat	
Software	
Data	

A2L

Environment - DefaultEnv

TBC mapping ID:

Customer_ID	default
ECU_ID	default
ECU Access	

Customer_ID

Fig. 4-5 The "UuT List" Tab of the Main Workspace

The list in the "UuT List" tab contains all UuTs of the UuT Group selected in the Project Explorer. A UuT in this list is characterized by its name ("UuT" column), the "UuT Description" ("Description" column) and the "UuT Environment" ("Environment" column).

The latter two are displayed in detail in the "Description" and "Environment" fields to the right of the main workspace.

Note

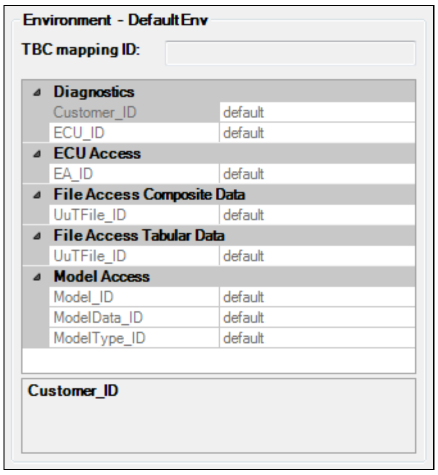
The assignment of a description and an environment to a UuT is defined in the UuT List and can only be modified there (see "The UuT List Editor" on page 186).

The "Description" Field

With a UuT of the type "ECU", this field contains a "Device" which in turn can contain any number of processors. Within this hierarchy, program and other data for this ECU are specified.

The "Environment" Field

This field mainly displays the "TBC-Mapping ID" and a range of what are referred to as "ID strings" which are of significance for the configuration of the test bench for the selected UuT.



Environment - DefaultEnv

TBC mapping ID:

Diagnostics	
Customer_ID	default
ECU_ID	default
ECU Access	
EA_ID	default
File Access Composite Data	
UuTFile_ID	default
File Access Tabular Data	
UuTFile_ID	default
Model Access	
Model_ID	default
ModelData_ID	default
ModelType_ID	default

Customer_ID

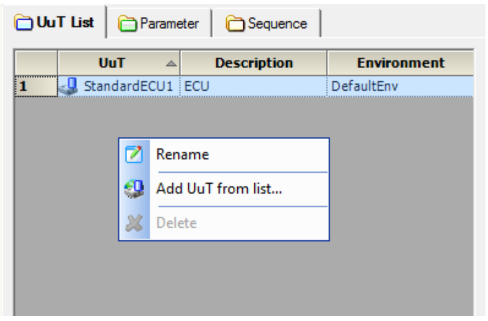
The values of these strings can be edited in the UuT List Editor (see "The UuT List Editor" on page 186) – e.g. when the settings provided with the "Global UuT List" do not correspond to the current tool configuration.

Note

The IDs specified here are resolved using the entries in the "Tool Configuration File".

To add a UuT to the UuT Group

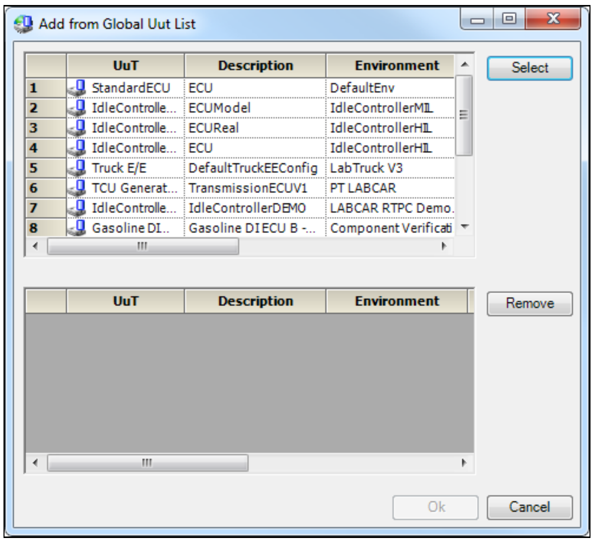
- Select a UuT Group in the Project Explorer.
- Select the "UuT List" tab in the main workspace.
- Select **Add UuT from list** from the shortcut menu.



	UuT	Description	Environment
1	StandardECU1	ECU	DefaultEnv

The "Add from Global UuT List" window opens.

- Select one or more UuTs from this list.



	UuT	Description	Environment
1	StandardECU	ECU	DefaultEnv
2	IdleControlle...	ECUModel	IdleControllerMIL
3	IdleControlle...	ECUReal	IdleControllerHIL
4	IdleControlle...	ECU	IdleControllerHIL
5	Truck E/E	DefaultTruckEEConfig	LabTruck V3
6	TCU Generat...	TransmissionECUV1	PT LABCAR
7	IdleControlle...	IdleControllerDEMO	LABCAR RTPC Demo.
8	Gasoline DI...	Gasoline DI ECU B -...	Component Verificati

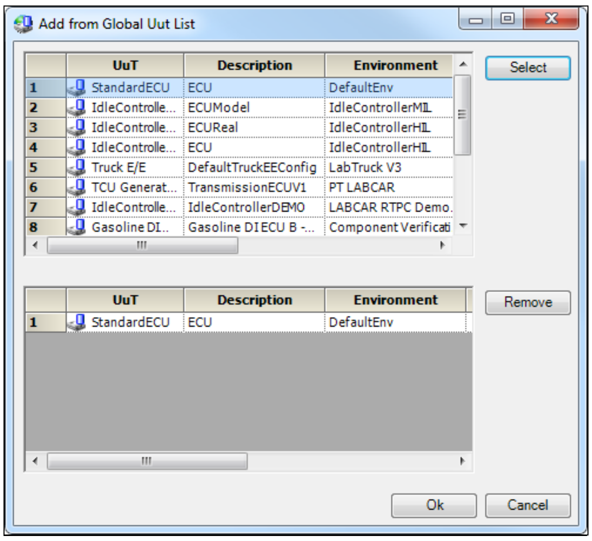
UuT	Description	Environment

Note

The directory containing your Global UuT List can be determined by the Test Project Manager – for more details see "To select a Global UuT List" on page 59.

- Click **Select**.

  The selected UuT(s) are added to the list in the bottom part of the window.

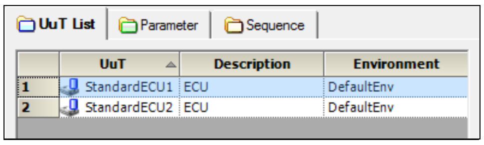


	UuT	Description	Environment
1	StandardECU	ECU	DefaultEnv
2	IdleControlle...	ECUModel	IdleControllerMIL
3	IdleControlle...	ECUReal	IdleControllerHIL
4	IdleControlle...	ECU	IdleControllerHIL
5	Truck E/E	DefaultTruckEEConfig	LabTruck V3
6	TCU Generat...	TransmissionECUV1	PT LABCAR
7	IdleControlle...	IdleControllerDEMO	LABCAR RTPC Demo.
8	Gasoline DI...	Gasoline DI ECU B -...	Component Verificati

	UuT	Description	Environment
1	StandardECU	ECU	DefaultEnv

If you have selected too many UuTs from an extensive "Global UuT List", you can remove these from the list of chosen UuTs by selecting them and clicking **Remove**.

- Click **OK** to add the desired UuTs to your UuT Group.

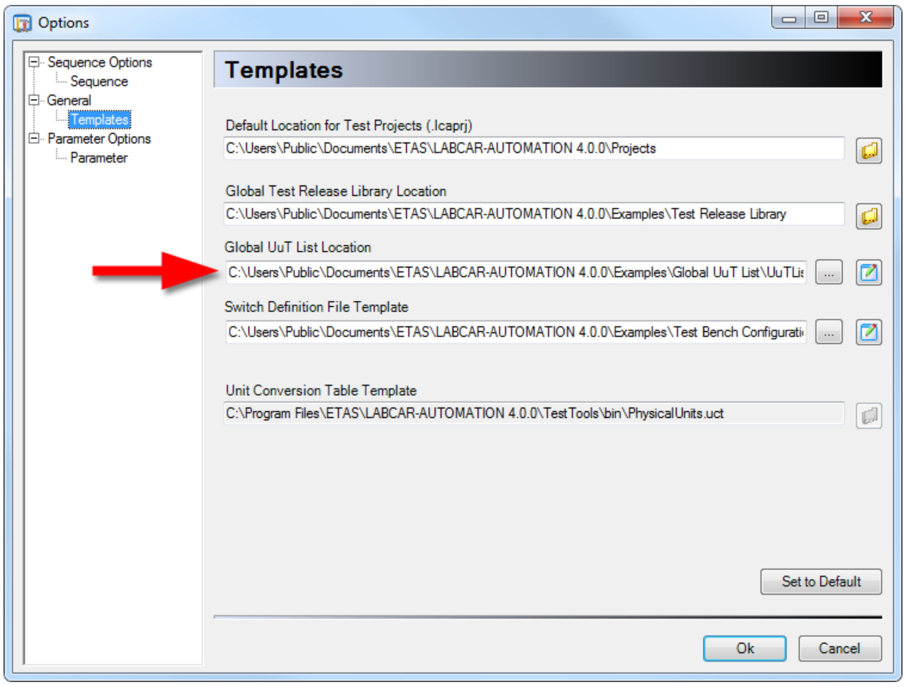


	UuT	Description	Environment
1	StandardECU1	ECU	DefaultEnv
2	StandardECU2	ECU	DefaultEnv

To select a Global UuT List

- Select **Tools** → **Options** in the main menu of Test Manager.

  The "Options" window opens.



Default Location for Test Projects (.lcaprj)
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Projects

Global Test Release Library Location
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Release Library

Global UuT List Location
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Global UuT List\UuTLis

Switch Definition File Template
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Bench Configurati

Unit Conversion Table Template
C:\Program Files\ETAS\LABCAR-AUTOMATION 4.0.0\TestTools\bin\PhysicalUnits.uct

- Click the folder icon next to "Global UuT List Location".

  A file selector window opens.

- Select the file which contains your Global UuT List (e.g. `UuTList.dbu`).

- To open the selected file in the UuT List Editor, click the editor icon.

For details of how to create a Global UuT List, refer to the section "The UuT List Editor" on page 186 in the tutorial.

To rename a UuT

- Select the ”UuT List” tab.
- Select the UuT to be renamed from the list.
- Select **Rename** from the shortcut menu.

  The ”(UuT) Editor” window opens.
- Enter a different name under ”Name” and, if necessary, change the comment under ”Comment”.

To delete a UuT from the UuT Group

- Select the ”UuT List” tab.
- Select the UuT to be deleted from the list.
- Select **Delete** from the shortcut menu.

  The selected UuT is deleted from the UuT Group.

4.1.3 Adding Test Functionality

Once the test project and UuT Groups have been created and UuTs have been assigned to them, the TPrM assigns test functionality to these UuT Groups.

This takes place in the Functionality Browser – the functionality assigned to a UuT Group is displayed here and managed by the TPrM.

The following Fig. 4-6 shows the Functionality Browser for the UuT Group ”ICECU008” selected in the Project Explorer – there is no functionality (i.e. test cases) to date.

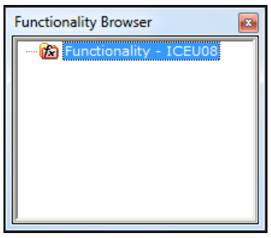


Fig. 4-6 The Functionality Browser

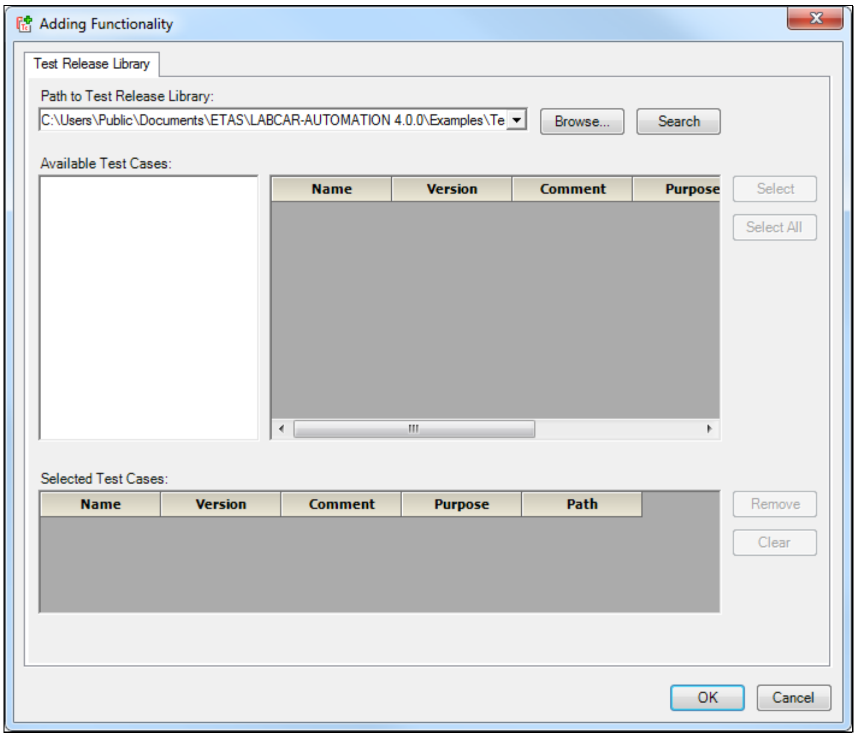
To assign test functionality to this UuT Group, proceed as follows:

To assign test functionality to a UuT Group

- Select the UuT Group to which you want to add functionality in the Project Explorer.
- Select the entry "Functionality - <*UuT Group name*>" in the Functionality Browser and press the right-hand mouse button.

- Select **Add Functionality** from the shortcut menu.

  The "Adding Functionality" window opens.

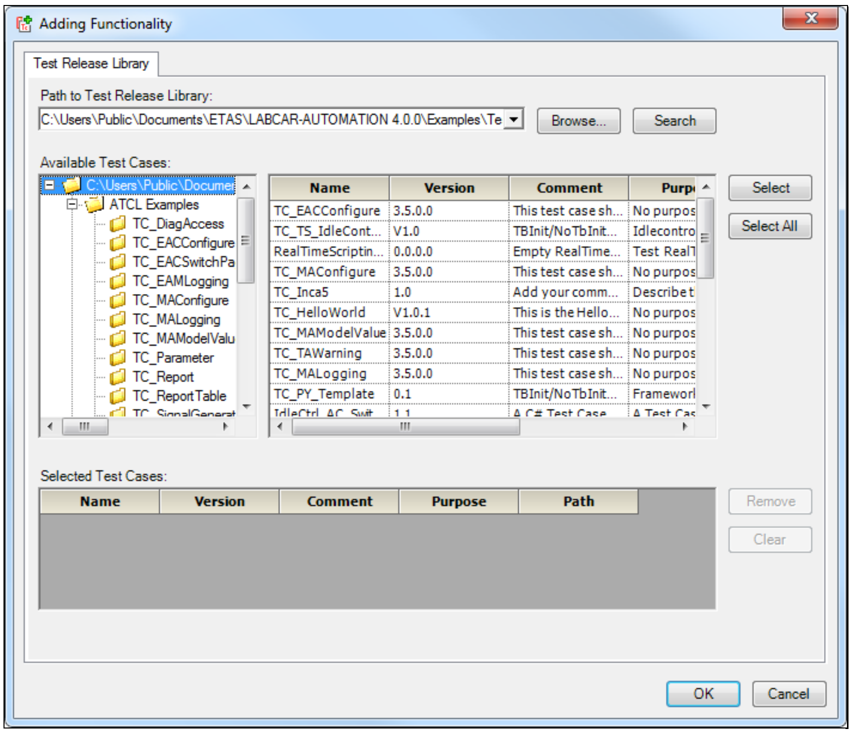


Note

Make sure that the path to your Test Case Library is set correctly (see "To determine a Test Case Library" on page 64).

- Click **Search**.

  The Test Case Library is searched and the test cases found shown in the "Available Test Cases" field.

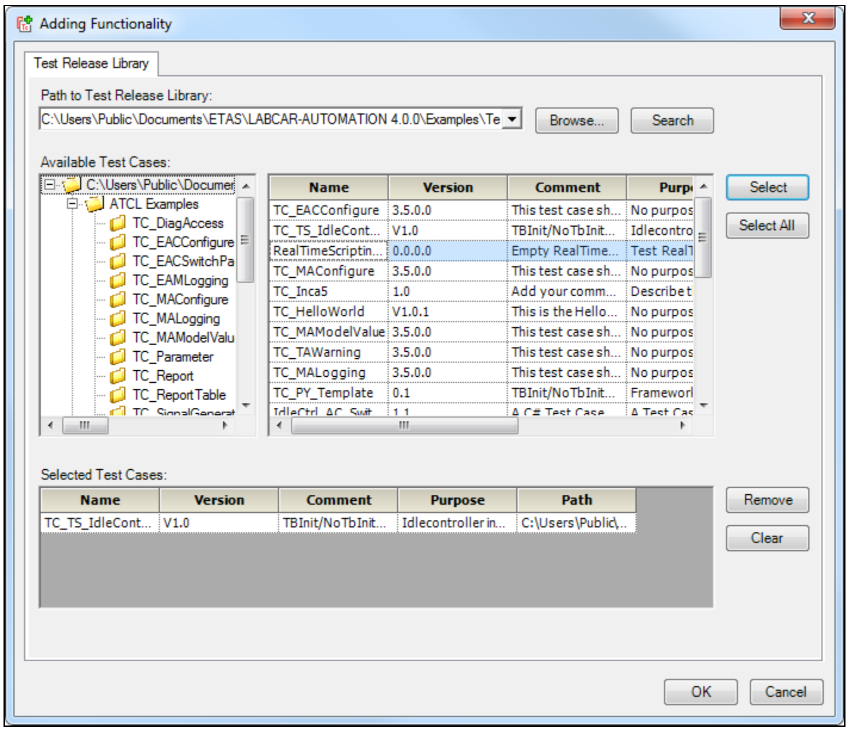


Name	Version	Comment	Purp
TC_EACConfigure	3.5.0.0	This test case sh...	No purpos
TC_TS_IdleCont...	V1.0	TBInit/NoTbInit...	Idlecontro
RealTimeScriptin...	0.0.0.0	Empty RealTime...	Test RealT
TC_MAConfigure	3.5.0.0	This test case sh...	No purpos
TC_Inca5	1.0	Add your comm...	Describe t
TC_HelloWorld	V1.0.1	This is the Hello...	No purpos
TC_MAModelValue	3.5.0.0	This test case sh...	No purpos
TC_TAWarning	3.5.0.0	This test case sh...	No purpos
TC_MALogging	3.5.0.0	This test case sh...	No purpos
TC_PY_Template	0.1	TBInit/NoTbInit...	Framewor
IdleCtrl_AC_Swit	1.1	A C# Test Case	A Test Cas

Name	Version	Comment	Purpose	Path

- Select one test case after the other by double-clicking or using **Select**.

  The selected test cases are added in the "Selected Test Cases" field.



Path to Test Release Library:

C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Te

Available Test Cases:

- C:\Users\Public\Documer
  - ATCL Examples
    - TC_DiagAccess
    - TC_EACConfigure
    - TC_EACSwitchPa
    - TC_EAMLogging
    - TC_MAConfigure
    - TC_MALogging
    - TC_MAModelValu
    - TC_Parameter
    - TC_Report
    - TC_ReportTable

Name	Version	Comment	Purp
TC_EACConfigure	3.5.0.0	This test case sh...	No purpos
TC_TS_IdleCont...	V1.0	TBInit/NoTbInit...	Idlecontro
RealTimeScriptin...	0.0.0.0	Empty RealTime...	Test RealT
TC_MAConfigure	3.5.0.0	This test case sh...	No purpos
TC_Inca5	1.0	Add your comm...	Describe t
TC_HelloWorld	V1.0.1	This is the Hello...	No purpos
TC_MAModelValue	3.5.0.0	This test case sh...	No purpos
TC_TAWarning	3.5.0.0	This test case sh...	No purpos
TC_MALogging	3.5.0.0	This test case sh...	No purpos
TC_PY_Template	0.1	TBInit/NoTbInit...	Framewor

Selected Test Cases:

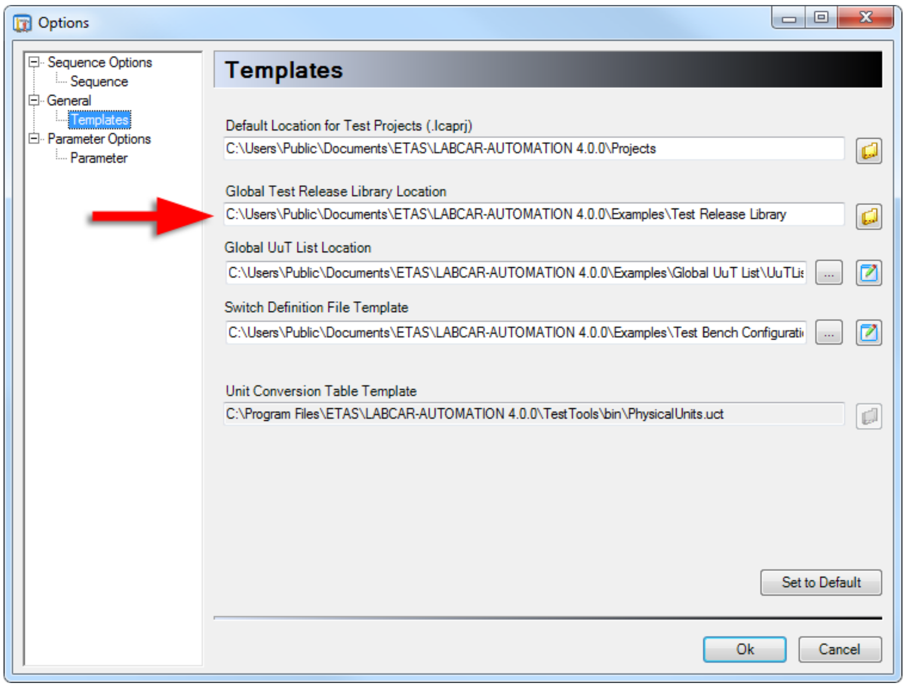
Name	Version	Comment	Purpose	Path
TC_TS_IdleCont...	V1.0	TBInit/NoTbInit...	Idlecontroller in...	C:\Users\Public\...

- To remove an entry from the list of available test cases select it and click **Remove**.
- To remove all test cases selected so far from the list, select **Clear**.
- To assign the test cases selected previously to the UuT Group, click **OK**.

To determine a Test Case Library

- Select **Tools** → **Options** in the main menu of Test Manager.

  The "Options" window opens.



Default Location for Test Projects (.lcaprj)
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Projects

Global Test Release Library Location
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Release Library

Global UuT List Location
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Global UuT List\UuTLis

Switch Definition File Template
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Bench Configurati

Unit Conversion Table Template
C:\Program Files\ETAS\LABCAR-AUTOMATION 4.0.0\TestTools\bin\PhysicalUnits.uct

- Click the folder icon next to "Global Test Release Library Location".

  A file selector window opens.

- Select the directory containing your Global Test Release Library.

To rename test functionality

- Select the test case to be renamed in the Functionality Browser.
- Select **Rename** from the shortcut menu.
- The "(Functionality Usage) Property Editor" window opens.
- Change the name ("Name") and, if necessary, the comment ("Comment").

To duplicate test functionality

- Select the test case to be duplicated in the Functionality Browser.

- Select **Duplicate Test Case** from the shortcut menu.

  A copy of the selected test case is created.

To delete test functionality from the UuT Group

- Select the test case to be deleted from the Functionality Browser.
- Select **Delete** from the shortcut menu.

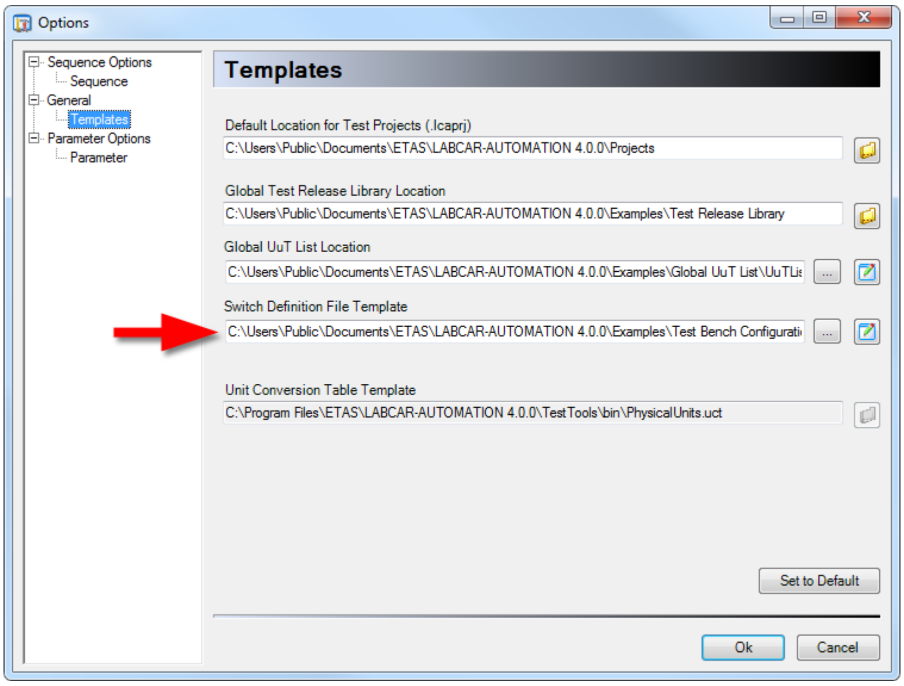
  The selected test case is deleted.

4.1.4 Selection of the Switch Definition File

The Test Project Manager selects the Switch Definition File to be used for the current project. Switch Definition Files are created during test development and saved as XML files (*.lcasdt). For more information on this topic, refer to the sections "The Switch Definition File" on page 183 and "The Switch Definition File Editor" on page 174.

To select the Switch Definition File

- Select **Tools** → **Options** in the main menu of Test Manager.
- The "Options" window opens.



Default Location for Test Projects (.lcaprj)
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Projects

Global Test Release Library Location
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Release Library

Global UuT List Location
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Global UuT List\UuTLis

Switch Definition File Template
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Bench Configurati

Unit Conversion Table Template
C:\Program Files\ETAS\LABCAR-AUTOMATION 4.0.0\TestTools\bin\PhysicalUnits.uct

- Click the folder icon next to "Switch Definition File Template".

  A file selector window opens.

- Select the file which contains the switch definition (e.g. `SwitchDefinitionTable.lcasdt`).
- To open the selected file in the Switch Definition File Editor, click the editor icon.

4.1.5 Synchronization

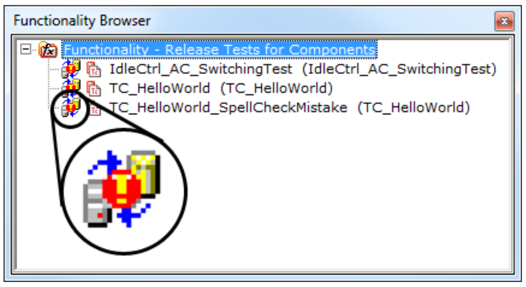
Synchronizing, as far as possible, is the automated comparing and updating of local data with global data.

A lot of the data used in the Test Manager (e.g. test cases with parameter structure, UuTs etc.) is copied into the test projects from global data structures. For example, a test case from the global Test Release Library is copied into the project's local library when selected in the Test Manager. This makes it possible for a test project to be transferred completely from one system to another by copying a local directory.

It also means that the test project can be run independently even if there is no network connection (or if the network fails) without depending on data from the global center.

If a Test Case Developer creates a new version of a test case and places it in the global library, the version in the global library differs from that in a test project using the test case. The same is true of the UuT and the switch definition.

This discrepancy between project and global library is indicated in the Test Manager by an icon to the left of the test case in the Functionality Browser.


Functionality Browser

Functionality - Release Tests for Components
- IdleCtrl_AC_SwitchingTest (IdleCtrl_AC_SwitchingTest)
- TC_HelloWorld (TC_HelloWorld)
- TC_HelloWorld_SpellCheckMistake (TC_HelloWorld)

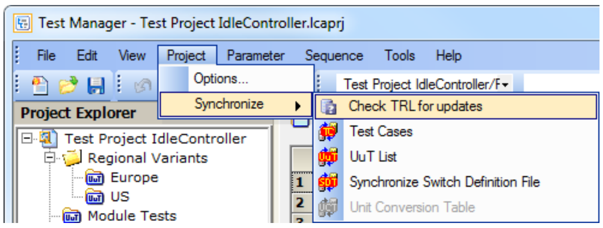
The Project Manager can now decide whether to synchronize automatically; this means that LABCAR-AUTOMATION checks whether it can accept the new test case in the project because test case structure and particularly parameter structure are sufficiently similar.

If so, the new test case is accepted, all sequences remain the same and tests already assigned data are accepted.

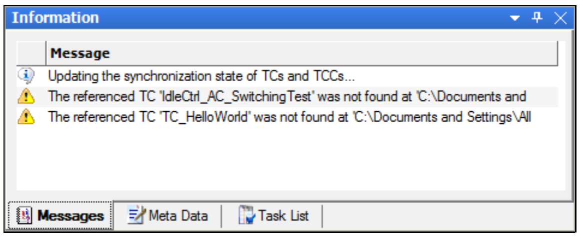
The algorithms for checking similarity are limited to the values and sequences stored in the test case. Checking is therefore limited (e.g. with obvious contradictions).

To check if the test cases are up-to-date

- Select **Project** → **Synchronize** → **Check TRL for updates**.



If changes have been made to the test cases used, an appropriate message is issued in the "Messages" tab in the "Information" window.

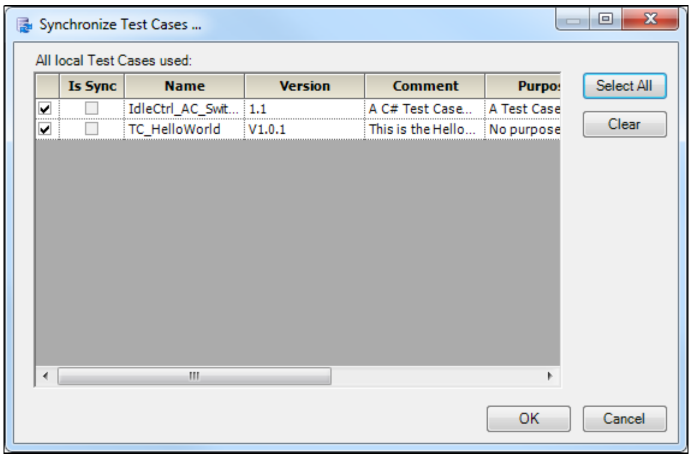


Message
Updating the synchronization state of TCs and TCCs...
The referenced TC 'IdleCtrl_AC_SwitchingTest' was not found at 'C:\Documents and
The referenced TC 'TC_HelloWorld' was not found at 'C:\Documents and Settings\All

To synchronize test cases

- Select **Project** → **Synchronize** → **Test Cases**.

The "Synchronize Test Cases ..." window opens.



All local Test Cases used:

	Is Sync	Name	Version	Comment	Purpo:
✓	☐	IdleCtrl_AC_Swit...	1.1	A C# Test Case...	A Test Case
✓	☐	TC_HelloWorld	V1.0.1	This is the Hello...	No purpose

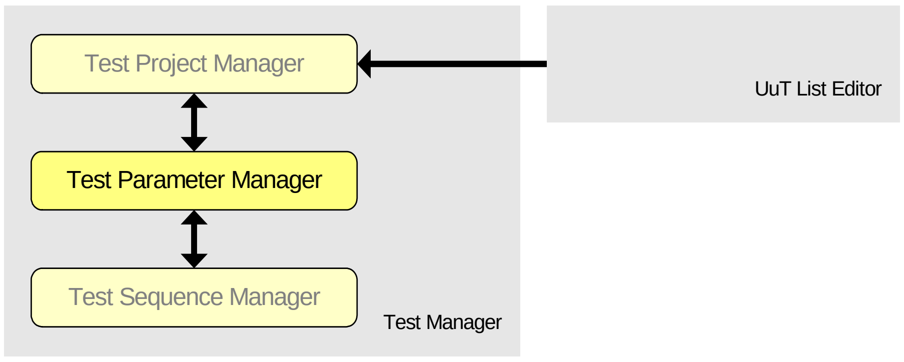
If there is no tick in the column "Is Sync", the test case used is not identical to the "Master" in the TRL.

- Click **Select All**.
- Click **OK**.
- The selected test cases are synchronized.

As described above, UuT Lists and Switch Definition Files can also be synchronized in this way.

4.2 The Tasks of the Test Parameter Manager

The Test Parameter Manager (TPM) transfers a logical test project (defined by the Test Project Manager) into a physical test project. This particularly means that it assigns all parameters (defined by the Test Case Developer) suitable values or releases the default values assigned by the Test Case Developer.



The TPM has the following tasks:

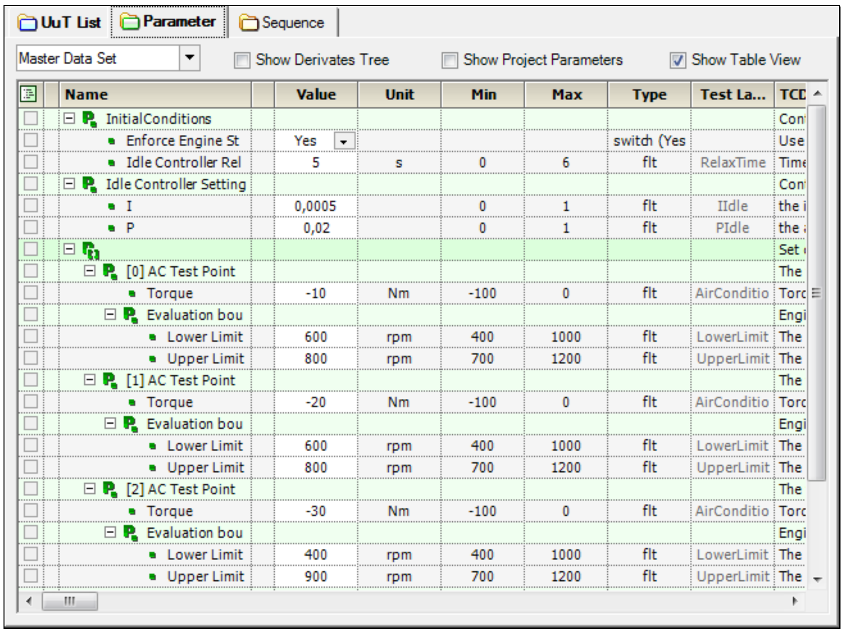
- Value assignment for all parameters of the test cases
- Definition of global project parameters and of Value Constants

Below, you will find information on executing the different tasks of the TPM. The individual sections are:

- "Selecting Parameters" on page 69
- "The Parameter Table" on page 70
- "The Shortcut Menu of the Parameter Table" on page 75
- "Editing Parameter Lists" on page 77
- "The Search Function" on page 79
- "Working with Project Parameters" on page 79
- "Working with Value Constants" on page 81
- "Creating Derivates" on page 84
- "Exporting and Importing Parameter Sets" on page 88
- "Options" on page 90

4.2.1 Selecting Parameters

The "Parameter" tab of the main workspace is available to the TPM for parameterization (Fig. 4-7). The content of this tab consists of a list, the lines of which consist of parameters – the parameter attributes are listed in the columns.



Name	Value	Unit	Min	Max	Type	Test La...	TCD
InitialConditions							Con
Enforce Engine St	Yes				switch (Yes		Use
Idle Controller Rel	5	s	0	6	flt	RelaxTime	Time
Idle Controller Setting							Con
I	0,0005		0	1	flt	IIdle	the i
P	0,02		0	1	flt	PIdle	the a
							Set
[0] AC Test Point							The
Torque	-10	Nm	-100	0	flt	AirConditio	Torc
Evaluation bou							Engi
Lower Limit	600	rpm	400	1000	flt	LowerLimit	The
Upper Limit	800	rpm	700	1200	flt	UpperLimit	The
[1] AC Test Point							The
Torque	-20	Nm	-100	0	flt	AirConditio	Torc
Evaluation bou							Engi
Lower Limit	600	rpm	400	1000	flt	LowerLimit	The
Upper Limit	800	rpm	700	1200	flt	UpperLimit	The
[2] AC Test Point							The
Torque	-30	Nm	-100	0	flt	AirConditio	Torc
Evaluation bou							Engi
Lower Limit	400	rpm	400	1000	flt	LowerLimit	The
Upper Limit	900	rpm	700	1200	flt	UpperLimit	The

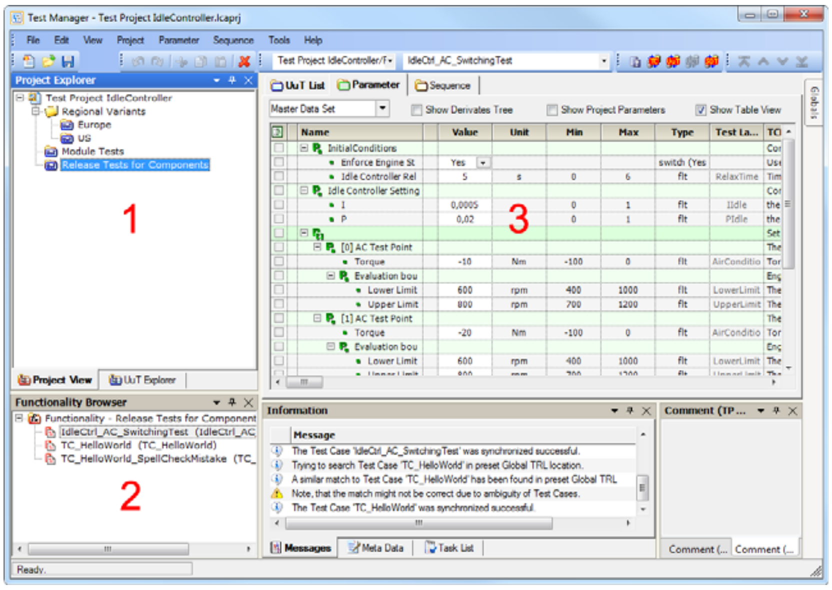
Fig. 4-7 The "Parameter" Tab of the Main Workspace

The parameters displayed in the list depend on the UuT Group selected in the Project Explorer and the test case selected in the Functionality Browser.

The sequence is as follows (see Fig. 4-8):

1. A UuT Group is selected in the Project Explorer. Alternatively, you can select a UuT in the "UuT Explorer" tab which means the relevant UuT Group is selected automatically.
2. The available functionality of the selected hierarchy level or the selected UuT Group is then displayed in the Functionality Browser.

3. All parameters belonging to the selected functionality are then displayed in the ”Parameter” tab. The displaying order is as determined by the Test Case Developer.


Test Manager - Test Project IdleController.lcaprj
File Edit View Project Parameter Sequence Tools Help
Test Project IdleController/T
IdleCtrl_AC_SwitchingTest
Project Explorer
Test Project IdleController
Regional Variants
Europe
US
Module Tests
Release Tests for Components
1
Project View
UuT Explorer
Functionality Browser
Functionality - Release Tests for Component
IdleCtrl_AC_SwitchingTest (IdleCtrl_AC
TC_HelloWorld (TC_HelloWorld)
TC_HelloWorld_SpellCheckMistake (TC_
2
UuT List
Parameter
Sequence
Master Data Set
Show Derivates Tree
Show Project Parameters
Show Table View
3
Globals
Information
Comment (TP...
Messages
Meta Data
Task List
Comment (...
Comment (...
Ready.

Name	Value	Unit	Min	Max	Type	Test La...	TC
InitialConditions							Cor
Enforce Engine St	Yes				switch (Yes		Use
Idle Controller Rel	5	s	0	6	flt	RelaxTime	Tim
Idle Controller Setting							Cor
I	0,0005		0	1	flt	IIdle	the
P	0,02		0	1	flt	PIdle	the
							Set
[0] AC Test Point							The
Torque	-10	Nm	-100	0	flt	AirConditio	Tor
Evaluation bou							Eng
Lower Limit	600	rpm	400	1000	flt	LowerLimit	The
Upper Limit	800	rpm	700	1200	flt	UpperLimit	The
[1] AC Test Point							The
Torque	-20	Nm	-100	0	flt	AirConditio	Tor
Evaluation bou							Eng
Lower Limit	600	rpm	400	1000	flt	LowerLimit	The

Message
The Test Case 'IdleCtrl_AC_SwitchingTest' was synchronized successful.
Trying to search Test Case 'TC_HelloWorld' in preset Global TRL location.
A similar match to Test Case 'TC_HelloWorld' has been found in preset Global TRL
Note, that the match might not be correct due to ambiguity of Test Cases.
The Test Case 'TC_HelloWorld' was synchronized successful.

Fig. 4-8 Steps in Parameter Selection (see Text)

4.2.2 The Parameter Table

The parameters and their attributes which belong to the selected functionality are displayed in the parameter table.

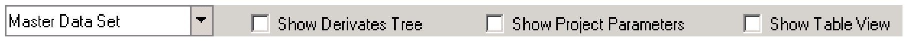
The inner elements of user records and dynamic arrays are not automatically displayed in this list to make sure that the view is clear.

User records and dynamic arrays are displayed with icons – dynamic arrays are also shown on a green background.

- LCA Dynamic Array
- LCA User Record

Further Display Options for the Parameter List

Above the parameter list there are three boxes with which the content and representation of the list can be influenced.



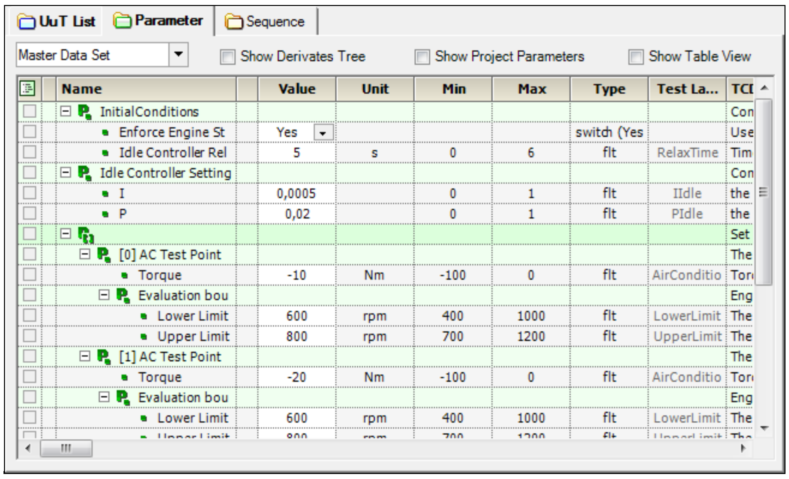
- Show Derivates Tree

  If this option is active, a field is displayed to the left of the parameter list in which master data set, derivates and UuTs assigned to them are displayed (for more details see "Creating Derivates" on page 84). As this field requires space to display the parameter list, existing derivates can be selected from the list to the left.

- Show Project Parameters

  If this option is active, only project parameters are shown in the parameter list (see "Working with Project Parameters" on page 79).
- Show Table View

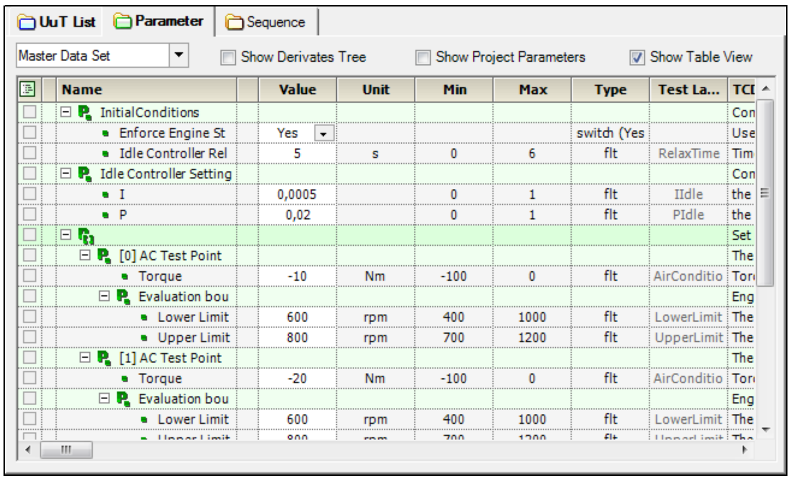
  This makes the display of dynamic arrays clearer.


UuT List | Parameter | Sequence

Master Data Set ☐ Show Derivates Tree ☐ Show Project Parameters ☐ Show Table View

Name	Value	Unit	Min	Max	Type	Test La...	TCI
⊟ InitialConditions							Con
Enforce Engine St	Yes				switch (Yes		Use
Idle Controller Rel	5	s	0	6	flt	RelaxTime	Tim
⊟ Idle Controller Setting							Con
I	0,0005		0	1	flt	IIdle	the
P	0,02		0	1	flt	PIdle	the
⊟							Set
⊟ [0] AC Test Point							The
Torque	-10	Nm	-100	0	flt	AirConditio	Tor
⊟ Evaluation bou							Eng
Lower Limit	600	rpm	400	1000	flt	LowerLimit	The
Upper Limit	800	rpm	700	1200	flt	UpperLimit	The
⊟ [1] AC Test Point							The
Torque	-20	Nm	-100	0	flt	AirConditio	Tor
⊟ Evaluation bou							Eng
Lower Limit	600	rpm	400	1000	flt	LowerLimit	The

Fig. 4-9 Normal Display of a Dynamic Array


UuT List | Parameter | Sequence

Master Data Set ☐ Show Derivates Tree ☐ Show Project Parameters ☑ Show Table View

Name	Value	Unit	Min	Max	Type	Test La...	TCI
⊟ InitialConditions							Con
Enforce Engine St	Yes				switch (Yes		Use
Idle Controller Rel	5	s	0	6	flt	RelaxTime	Tim
⊟ Idle Controller Setting							Con
I	0,0005		0	1	flt	IIdle	the
P	0,02		0	1	flt	PIdle	the
⊟							Set
⊟ [0] AC Test Point							The
Torque	-10	Nm	-100	0	flt	AirConditio	Tor
⊟ Evaluation bou							Eng
Lower Limit	600	rpm	400	1000	flt	LowerLimit	The
Upper Limit	800	rpm	700	1200	flt	UpperLimit	The
⊟ [1] AC Test Point							The
Torque	-20	Nm	-100	0	flt	AirConditio	Tor
⊟ Evaluation bou							Eng
Lower Limit	600	rpm	400	1000	flt	LowerLimit	The

Fig. 4-10 Display with the Option "Show Table View"

Sequence of the Elements of a Dynamic Array

As the sequence of the elements of a dynamic array has an effect on how a test case is run, it can be changed in the parameter list. Click the element to be moved and drag it to the required position within the dynamic array keeping the mouse button pressed down.

Scope of the Parameter List Attributes Displayed

The number of attributes displayed (i.e. the number of columns) can be selected by the user in different degrees.

To modify the display of the parameter table

- Right-click the head of the parameter table.

  A shortcut menu opens.
- Select **Optimal Column Width** to adapt the column width to suit the content of the columns.
- Select **Show Basic Attributes** to show only the elementary attributes (marked in Tab. 4-1 on page 74 with **).
- Select **Show Standard Attributes** to show only the standard attributes (marked in Tab. 4-1 on page 74 with *).
- Select **Show All Attributes** to show all parameter attributes.
- Select **Expand All** to display all lines of the parameter table.
- Select **Collapse All** to show user records and dynamic arrays only as one line each.

All attributes are described in the following table.

Column	Description		Editable by TPM?
TC Location	Test case in which the parameter is used		no
Name **	The name of the parameter object		only with a dynamic array
Column with symbols *		Parameter references a Value Constant	yes
		Parameter references a Project Parameter	
Value **	The value of the parameter. There are various ways of editing the parameter (depending on the type of parameter, see ”Type” below): - Scalar float/string: a simple text editor - Scalar bool: a checkbox which can be enabled (”True”) or disabled (”False”) - Scalar switch: a list which contains the possible switch positions - Scalar integer: a text box - All non-scalar parameters (apart from user records and dynamic arrays): The first value is displayed followed by ”...”. Double-clicking the cell opens a table editor. Note: *The editing mode can only be exited if the entry/change is consistent with the type of parameter and is within the defined limits.*		yes
Def Value	The default value		no
Unit **	The physical unit of the parameter as defined by the Test Case Developer. In the case of non-scalar parameters, the units of the values are specified. This column cannot be edited.		no
Min *	The lower limit of a numeric parameter – otherwise empty. This column cannot be edited.		no
Max *	The upper limit of a numeric parameter – otherwise empty. This column cannot be edited.		no

Column	Description	Editable by TPM?
Type *	The parameter type - Scalar parameters: ”int”, ”flt”, ”bool”, ”cstring”, ”hexstr”, ”octstr”, ”bitstr” - Scalar switches: "swi" , followed (in brackets) by the relevant switch type - Arrays: the data type followed by ”[]” - Look-up tables are referred to as ”Curve” (1-D) or ”Map” (2-D).	no
Test Label **	The parameter label which is used in the SUT Mapping File to access the test bench. This cell can be edited providing the Test Case Developer has allowed modifications.	yes
Default Test Label	The test label predefined by the Test Case Developer	no
Localized Name	Is not used at the moment	no
Instance Name	Name of the parameter in the Test Design Tool	no
TCD Comment **	A comment assigned by the Test Case Developer. See the ”Comment TCD” tab in the ”Comment” window.	no
TPM Comment **	The comment entered by the TPM in the ”Comment TPM” tab in the ”Comment” window.	yes
* Attribute of the type ”Standard” (see "To modify the display of the parameter table" on page 72) ** Attribute of the type ”Basic” (see "To modify the display of the parameter table" on page 72) By definition, all attributes of the type ”Basic” are also of the type ”Standard”.		

Tab. 4-1 Parameter Attributes

4.2.3 The Shortcut Menu of the Parameter Table

For working in the parameter table, there is a shortcut menu available to the Test Parameter Manager; this is described below. Non-trivial user actions are described in the following sections.

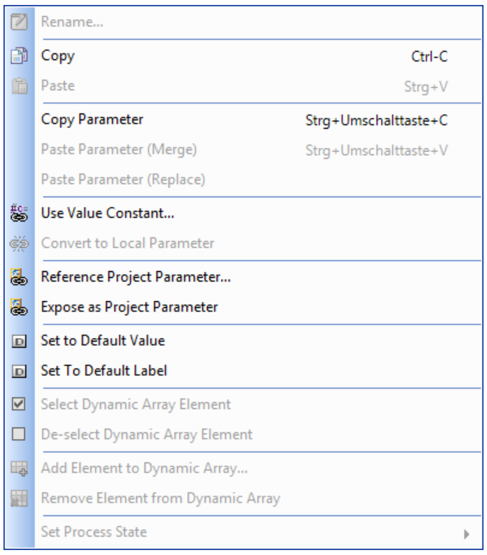

Rename...
Copy Ctrl-C
Paste Strg+V
Copy Parameter Strg+Umschalttaste+C
Paste Parameter (Merge) Strg+Umschalttaste+V
Paste Parameter (Replace)
Use Value Constant...
Convert to Local Parameter
Reference Project Parameter...
Expose as Project Parameter
Set to Default Value
Set To Default Label
Select Dynamic Array Element
De-select Dynamic Array Element
Add Element to Dynamic Array...
Remove Element from Dynamic Array
Set Process State

Fig. 4-11 The Shortcut Menu of the "Parameter" Tab

- **Rename**

  Renames elements of a dynamic array.

- **Copy**

  Copies the value of the selected parameter into the clipboard.

  Multiple selection is possible. If a non-scalar parameter has been selected, the entire array is copied into the clipboard.

- **Paste**

  Adds one or more parameters from the clipboard.

  The type and value range of the parameters or the access properties of labels which have to correspond to those in the clipboard may limit pasting.

  This menu item is disabled if the clipboard contains invalid objects. Referencing parameters are skipped during pasting.

- **Copy Parameter**

  Copies the content of a dynamic array to the clipboard which can be inserted into another dynamic array in two different ways (see below).

- **Paste Parameter (Merge)**

  Pastes the content of a dynamic array previously copied to the clipboard into another dynamic array. The following rules apply:

  – A parameter which only exists in the copied dynamic array is added to the new array.

  – A parameter which is both in the copied dynamic array and in the new array is overwritten with the value in the clipboard.

  – A parameter which only occurs in the new array is not changed.

- **Paste Parameter (Replace)**

  In this version of pasting, the content of the existing dynamic array is completely replaced by the content to be pasted.

- **Use Value Constant**

  Opens a dialog box listing all available Value Constants and from which one can be selected (see "To use a Value Constant" on page 83). The parameter value is then replaced by the constant value – in the ”Parameter” tab, the name of the constant is displayed instead of the value.

- **Convert to Local Parameter**

  This menu item is activated if the selected parameter references a Project Parameter. The reference is broken up and the parameter is defined locally.

- **Reference Project Parameter**

  References a Project Parameter (see next point).

- **Expose as Project Parameter**

  Makes the selected parameter a Project Parameter (which can be referenced).

- **Set to Default Value**

  Sets the parameter value to the default value specified by the Test Case Developer.

- **Set to Default Label**

  Resets the parameter label to the label specified by the Test Case Developer.

- **Select Dynamic Array Element**

  Selects a previously deselected (see below) dynamic array.

- **De-select Dynamic Array Element**

  Deselects individual parameters of a dynamic array (for more details refer to "To deselect elements of a dynamic array" on page 78).

- **Add Element to Dynamic Array**

  Adds a copy of the selected element to the current dynamic array (at the end of the list).

- **Remove Element From Dynamic Array**

  The selected element is removed from the list. At least one entry, however, must remain.

  This menu item is only active if an element of a dynamic array is selected in the parameter table.

4.2.4 Editing Parameter Lists

Below, you will find a description of a range of elementary user actions in the parameter list.

To change the value of a parameter

- To change the value of a parameter, double-click the relevant cell ("Value" column).

  The value can now be edited.

To add or remove parameters to a dynamic array

- If you want to add a parameter to a dynamic array, select **Add Element to Dynamic Array** from the shortcut menu.

  A dialog box opens in which you can enter the name of the new element and a comment on it.
- If you want to remove a parameter from a dynamic array, select **Remove Element from Dynamic Array** from the shortcut menu.

To move several elements within a dynamic array

An individual parameter can be moved within a dynamic array using drag-and-drop. To move several parameters at the same time, proceed as follows:

- Use the <SHIFT> key to highlight a contiguous area within the dynamic array.

or

- Use the <CTRL> key to highlight several individual parameters of the dynamic array.
- Move the marked parameters by keeping the mouse button pressed down within the dynamic array.

To copy and edit parameters of a dynamic array

- Select the dynamic array the elements of which you want to copy.
- Select **Copy Parameter** from the shortcut menu.

  The content of the dynamic array is copied to the clipboard; this can be pasted to another dynamic array in two different ways.
- To merge the content of the clipboard with the content of another dynamic array, select **Paste Parameter (Merge)** from the shortcut menu (see "Paste Parameter (Merge)" on page 75).

- To replace the content of the dynamic array completely with the content of the clipboard, select **Paste Parameter (Replace)** from the shortcut menu.

To deselect elements of a dynamic array

It is possible to deselect individual elements of a dynamic array which means that these elements are not part of the parameter set during test execution.

- Select the element required.
- Press the right-hand mouse button and select **Deselect Dynamic Array Element** from the shortcut menu.

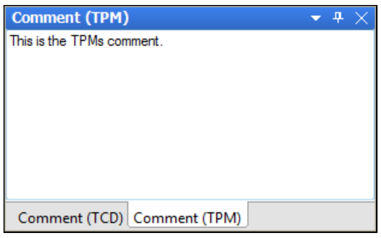
The relevant element is deselected (indicated by a red cross).

- If you no longer want these parameters to be displayed in the list, select **Tools** → **Options**
- Select the element "Parameter".
- Activate the option **Hide de-selected Parameters** under "Display of Dynamic Array Elements".

To document changes to parameters

In addition to the comments which can be assigned to a parameter by the Test Case Developer (TCD) and which can be read, but not edited, by all other roles in the test process, it is also possible for the Test Parameter Manager (TPM) to, for example, document changes to parameters using comments.

- Select the parameter to be commented.
- Select the "Comment (TPM)" tab in the "Comment" window.
- Enter a comment.


Comment (TPM)
This is the TPMs comment.

- Click outside the "Comment" window.

The comment is displayed in the "Comment (TPM)" column of the parameter list.

4.2.5 The Search Function

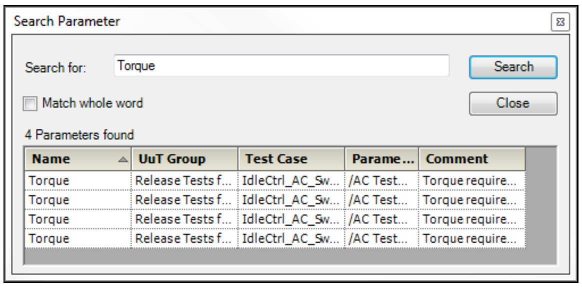
A search function is available to find individual parameters of a project.

To search for parameters

- In the main menu of Test Manager, select **Parameter** → **Search**.

  The "Search Parameter" window opens.
- Under "Search for:" enter a search string – please note the use of upper and lower case.
- If you only want to search for whole words, activate the "Match whole word" option.
- Click **Search**.

  The parameters found are listed in the window.

  Search for: Torque

  Match whole word

  4 Parameters found

Name	UuT Group	Test Case	Parame...	Comment
Torque	Release Tests f...	IdleCtrl_AC_Sw...	/AC Test...	Torque require...
Torque	Release Tests f...	IdleCtrl_AC_Sw...	/AC Test...	Torque require...
Torque	Release Tests f...	IdleCtrl_AC_Sw...	/AC Test...	Torque require...
Torque	Release Tests f...	IdleCtrl_AC_Sw...	/AC Test...	Torque require...

  For every parameter found, further information is displayed such as the UuT Group and the test case in which the parameter occurs, perhaps its location in a dynamic array or user record as well as the "Comment" attribute.
- Each of these columns can be sorted by double-clicking the column title to show in standard or reverse order.

4.2.6 Working with Project Parameters

Project Parameters are parameters which are available for referencing in the entire test project. This avoids giving "identical" parameters data on several occasions.

To make a parameter a Project Parameter, proceed as follows:

To define a Project Parameter

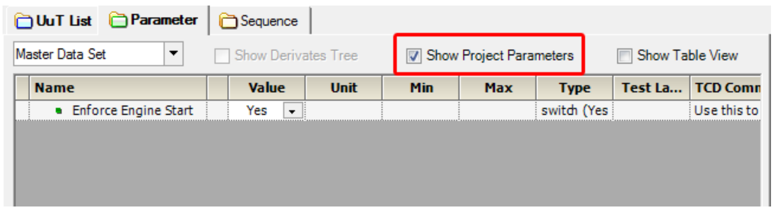
- Select **Expose as Project Parameter** from the shortcut menu of the parameter table.

  The parameter is added to the list of existing Project Parameters.

To display a Project Parameter

- To display the list of all existing Project Parameters, activate the "Show Project Parameter" option above the parameter table.

  The list of available Project Parameters is displayed.



Master Data Set

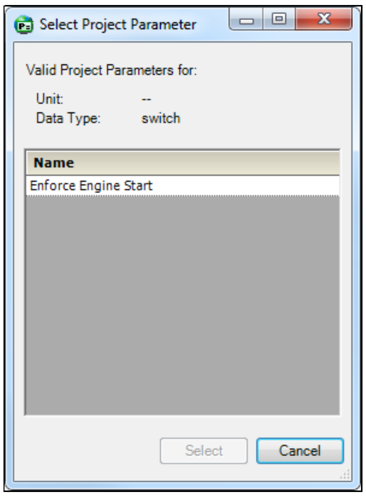
☑ Show Project Parameters

Name	Value	Unit	Min	Max	Type	Test La...	TCD Comn
Enforce Engine Start	Yes				switch (Yes		Use this to

To reference a Project Parameter

- Select the parameter which is to reference a Project Parameter.
- Select **Reference Project Parameter** from the shortcut menu.

  A window opens in which all available Project Parameters of the suitable data type are listed.



Valid Project Parameters for:

Unit:	--
Data Type:	switch

Name
Enforce Engine Start

- Select the required parameter from the list and click **Select**.

  This creates the referencing.

To cancel referencing

- Select the parameter whose referencing is to be canceled.

- Select **Convert to Local Parameter** from the shortcut menu.

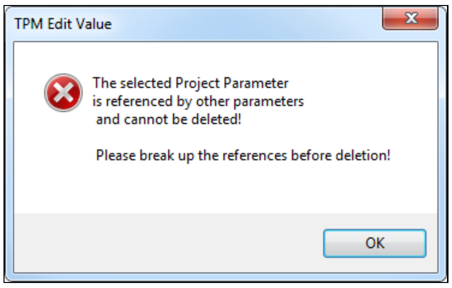
  This menu item is activated if the selected parameter references a Project Parameter or a Value Constant. The reference is broken up and the parameter is defined locally.

To delete a Project Parameter

- To delete a Project Parameter again, display the list of all available Project Parameters (see "To display a Project Parameter" on page 80).
- Select the parameter to be deleted.
- Select **Delete Project Parameter** from the shortcut menu.

  If the selected Project Parameter is not referenced by other parameters, it is deleted.

  Otherwise an error message is displayed.



The selected Project Parameter is referenced by other parameters and cannot be deleted!

Please break up the references before deletion!

- In this case, you have to break up the references before.

Note

The parameter itself is of course not deleted - just its characteristic of being a project parameter.

4.2.7 Working with Value Constants

The Test Parameter Manager can define constants for scalar values ("Value Constants") which can be used instead of a numeric value during parameterization. In this way, several parameters can reference the same value.

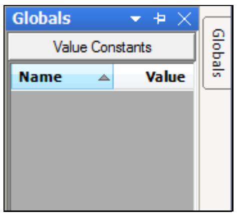
These constants are defined and edited in the "Globals" window under "Value Constants" (see "To define a Value Constant" on page 82). A "Value Constant" consists of a name ("Name"), type ("Type"), the assigned value ("Value"), its unit ("Unit") and a comment ("Comment").

The assignment of one of these constants to a parameter takes place via the shortcut menu of the parameter table (see "To use a Value Constant" on page 83). The prerequisite for successful value assignment is that the unit of the constant and the unit of the target parameter correspond to each other. If the value assignment is successful, the constant name and not the value of the constant is displayed in the relevant field of the parameter table.

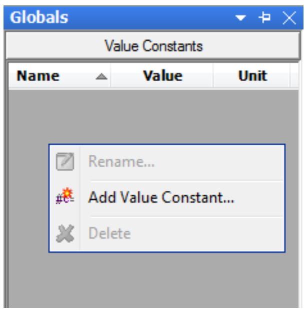
To define a Value Constant

To define a new Value Constant, proceed as follows:

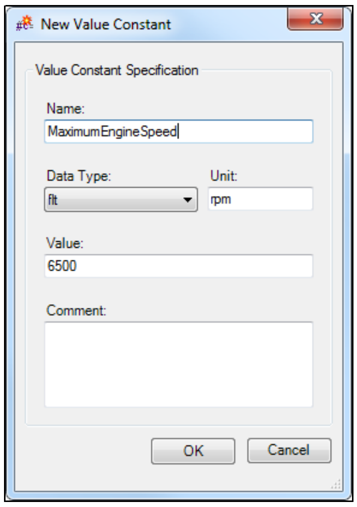
- Open the "Globals" window by moving the mouse pointer in Test Manager to the tab at the top right.



- Select **Add Value Constant** from the shortcut menu of this window.



The "New Value Constant" window opens.

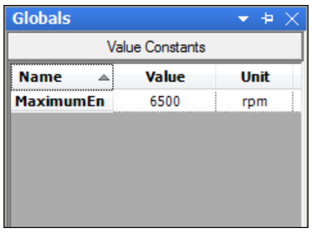


Name: MaximumEngineSpeed
Data Type: flt
Unit: rpm
Value: 6500
Comment:

- Enter a name.
- Select the data type and enter a unit and a comment.

- Press **OK**.

  The new Value Constant is added to the list and is now available for referencing with parameters.

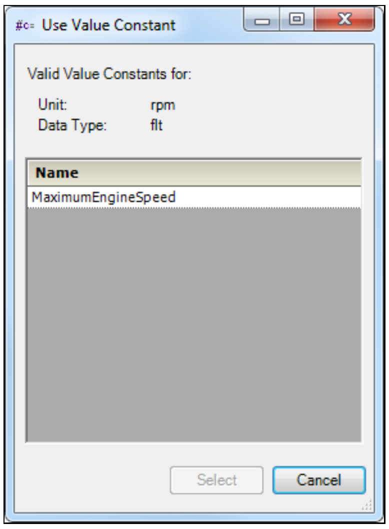


Name	Value	Unit
MaximumEn	6500	rpm

To use a Value Constant

- Select the parameter from the parameter table to which a Value Constant is to be assigned.
- Select **Use Value Constant** from the shortcut menu.

  The "Use Value Constant" window opens in which all Value Constants are listed whose unit and data type correspond to those of the parameter selected.



Valid Value Constants for:

Unit: rpm
Data Type: flt

Name
MaximumEngineSpeed

- Select the required Value Constant from the list and click **Select**.

  The parameter value is replaced by the constant value – in the parameter table, the name of the referenced Value Constant is displayed, not a numeric value, in the "Value" column.

To rename a Value Constant

- To rename a Value Constant, select it in the "Globals" window.
- Select **Rename** from the shortcut menu.

- Change the name and, if necessary, the comment.
- Click **OK**.

  The Value Constant is renamed.

To delete a Value Constant

- To delete a Value Constant, select it in the "Globals" window.
- Select **Delete** from the shortcut menu.

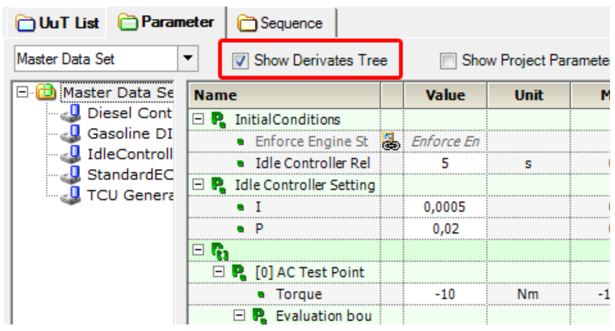
  If the selected Value Constant is not referenced by one or more parameters, it is deleted. Otherwise an error message is issued.

4.2.8 Creating Derivates

To be able to work with derivates (see section "Derivates" on page 21), you have to make them visible in the "Parameter" tab.

To display derivates

- Activate the "Show Derivates Tree" option in the "Parameter" tab.

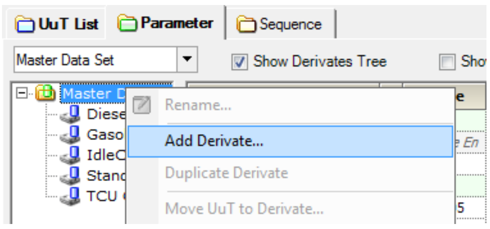


Name		Value	Unit	M
InitialConditions				
Enforce Engine St		Enforce En		
Idle Controller Rel		5	s	
Idle Controller Setting				
I		0,0005		
P		0,02		
[0] AC Test Point				
Torque		-10	Nm	-1
Evaluation bou				

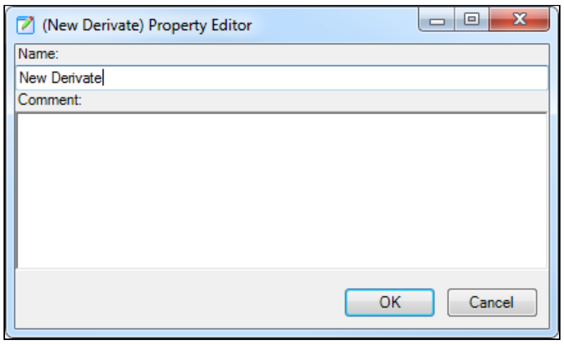
The master data set and the two UuTs of the UuT Group which work with this data set are shown to the left of the parameter list.

To create a derivate of the master data set

- Right-click the field showing the master data set.
- Select **Add Derivate**.

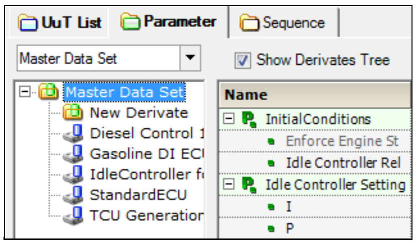


The "New Derivate" dialog box opens.



- Enter a name (and a comment if required).
- Click **OK**.

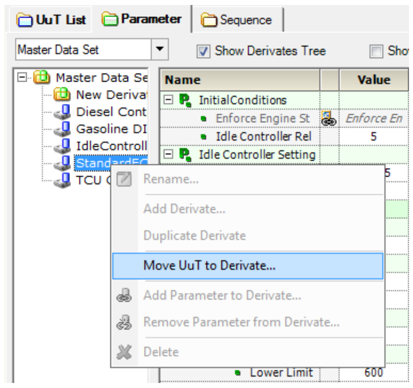
A new derivate is created.



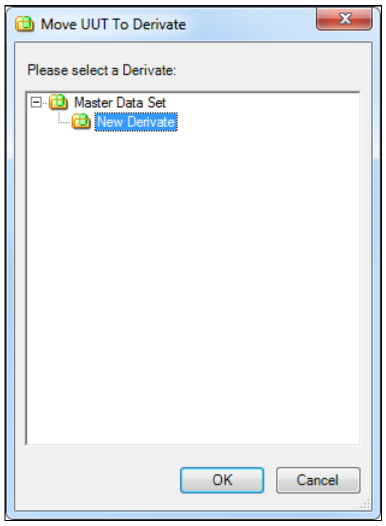
So far the derivate is "empty", i.e. it does not have any UuTs or parameters assigned to it yet (unlike the master data set).

To assign a UuT to a derivate

- Select the UuT to be assigned to the derivate.
- Select **Move UuT to Derivate** from the shortcut menu.

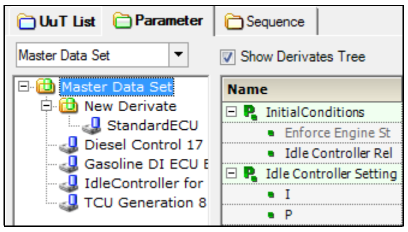


The "Move UuT to Derivate" window opens.



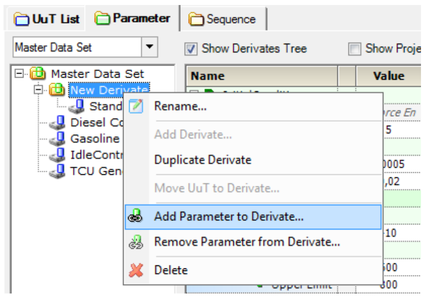
- Select the derivate to which you want to assign the UuT.
- Click **OK**.

The selected UuT is assigned to the selected derivate.



To assign parameters to the derivate

- Select the derivate you want to assign parameters to.
- Select **Add Parameter to Derivate** from the shortcut menu.

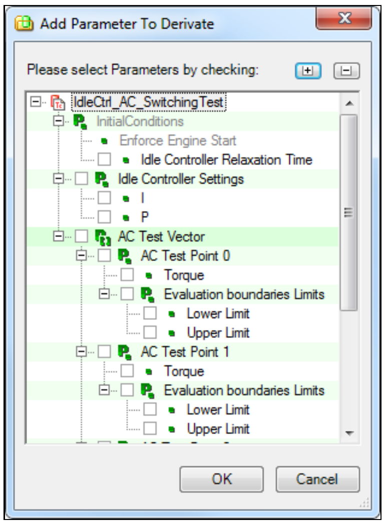


Note

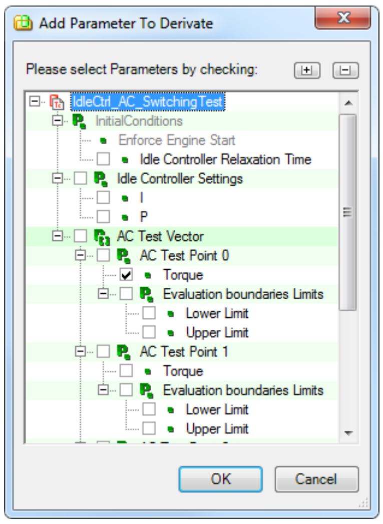
Make sure a test case from which the parameters are to be selected is activated in the Functionality Browser!

The "Add Parameter to Derivate" window opens.

- Click the "+" icon to show the hierarchy of the parameters of the selected test case in entirety.


Add Parameter To Derivate
Please select Parameters by checking:
IdleCtrl_AC_SwitchingTest
InitialConditions
Enforce Engine Start
Idle Controller Relaxation Time
Idle Controller Settings
I
P
AC Test Vector
AC Test Point 0
Torque
Evaluation boundaries Limits
Lower Limit
Upper Limit
AC Test Point 1
Torque
Evaluation boundaries Limits
Lower Limit
Upper Limit
OK
Cancel

- Select the desired parameter(s).



Add Parameter To Derivate

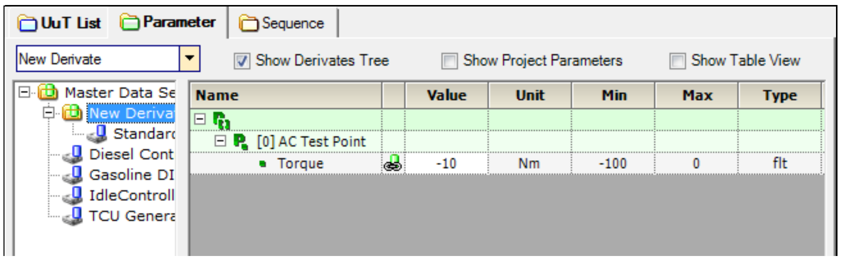
Please select Parameters by checking:

- IdleCtrl_AC_SwitchingTest
  - InitialConditions
    - Enforce Engine Start
    - [ ] Idle Controller Relaxation Time
  - [ ] Idle Controller Settings
    - [ ] I
    - [ ] P
  - [ ] AC Test Vector
    - [ ] AC Test Point 0
      - [✓] Torque
      - [ ] Evaluation boundaries Limits
        - [ ] Lower Limit
        - [ ] Upper Limit
    - [ ] AC Test Point 1
      - [ ] Torque
      - [ ] Evaluation boundaries Limits
        - [ ] Lower Limit
        - [ ] Upper Limit

OK Cancel

- Click **OK**.

  The parameter(s) is/are assigned to the derivate.



UuT List | Parameter | Sequence

New Derivate — [✓] Show Derivates Tree [ ] Show Project Parameters [ ] Show Table View

- Master Data Se
  - New Deriva
    - Standard
  - Diesel Cont
  - Gasoline DI
  - IdleControll
  - TCU Genera

Name	Value	Unit	Min	Max	Type
[0] AC Test Point					
Torque	-10	Nm	-100	0	flt

The icon after the “Name” column indicates that the parameter is part of a derivate.

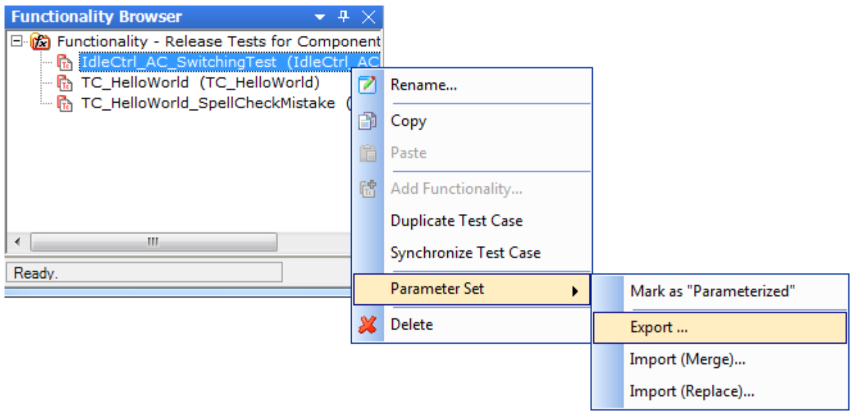
4.2.9 Exporting and Importing Parameter Sets

Parameterizations of a test case can be exported and reimported in entirety. To do this, proceed as follows:

To export parameters of a test case

- In the Functionality Browser select the test case whose parameters you would like to export.

- Select **Parameter Set** → **Export** from the shortcut menu.



The file selector window "Export Test Case Parameter Set" opens.

- Select a directory and enter a file name.
- Click **OK**.

The parameter set of the test case is saved as "LCA Parameter Set Exchange File" (file extension "lcapse").

To import parameters into a test case

- Select the test case into which you want to import parameters in the Functionality Browser.
- If you want to replace the existing parameters of dynamic arrays entirely by those of the parameter set to be imported, select **Parameter Set** → **Import (Replace)** from the shortcut menu.
- If you want to extend the existing parameters with those of the parameter set to be imported or overwrite them if they already exist, select **Parameter Set** → **Import (Merge)** from the shortcut menu.
- The "Import Test Case Parameter Set" file selector window opens.
- Select the file which contains the parameters to be imported.
- Click **OK**.

The parameters of the file are imported into the selected test case.

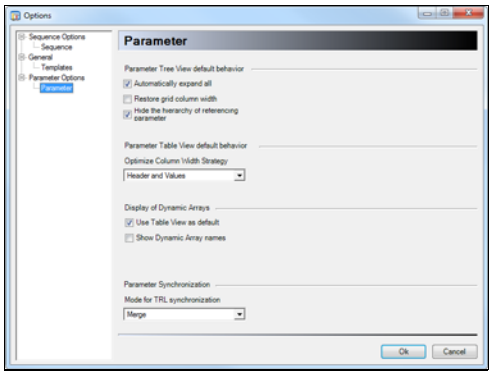
What happens to parameters which are parts of dynamic arrays is determined by the import variant selected. The following response is to be expected with parameters which do not belong to dynamic arrays:

- If a parameter of the imported parameter set is not part of the existing parameter set, it is not imported.
- If a parameter exists but has a different value, the existing value is overwritten by that of the imported parameter.

Messages on cases of a parameter being in the parameter set to be pasted but not in the target parameter set or vice versa are also issued in the "Task List" tab of the "Information" window (see "The "Task List" Tab" on page 44).

4.2.10 Options

There is a range of settings for displaying parameters in Test Manager. The relevant dialog box is invoked using **Tools** → **Options** (under "Parameter Options/ Parameter").



Parameter

Parameter Tree View default behavior
- [✓] Automatically expand all
- [ ] Restore grid column width
- [✓] Hide the hierarchy of referencing parameter

Parameter Table View default behavior
Optimize Column Width Strategy: Header and Values

Display of Dynamic Arrays
- [✓] Use Table View as default
- [ ] Show Dynamic Array names

Parameter Synchronization
Mode for TRL synchronization: Merge

Fig. 4-12 The "Options" Dialog Box: Parameters

Parameter Tree View Default Behavior

These settings concern the display of the parameter list in the "Parameter" tab.

- Automatically expand all

  The parameter list is shown in detail.
- Restore grid column width

  Activate this option if you would like to retain the column width of the parameter list when closing the project.
- Hide the hierarchy of referencing parameters

  This option results in the hierarchy of the referencing parameter no longer being displayed explicitly.

Parameter Table View Default Behavior

With this option, you can influence parameter representation (Table View):

- Optimize Column Width Strategy

  Adjusts the width of the column.

  – Header and Values

    Both the header and values in this column are taken into consideration for the column width.

  – Values only

    Only the values in this column are taken into consideration for the column width.

Display of Dynamic Arrays

These settings concern the display of "dynamic arrays".

- Hide de-selected Parameters

  Dynamic array parameters deselected temporarily are not displayed.

- Use Table View as default

  If this option is activated, "Table View" is selected as the default display (see "Show Table View" on page 71).

- Show Dynamic Array names

  If this option is activated, the names of dynamic arrays are also displayed in the parameter list.

Parameter Synchronization

This is where you can specify the procedure re the dynamic arrays of the test cases during synchronization:

- Mode for TRL synchronization

  – Merge

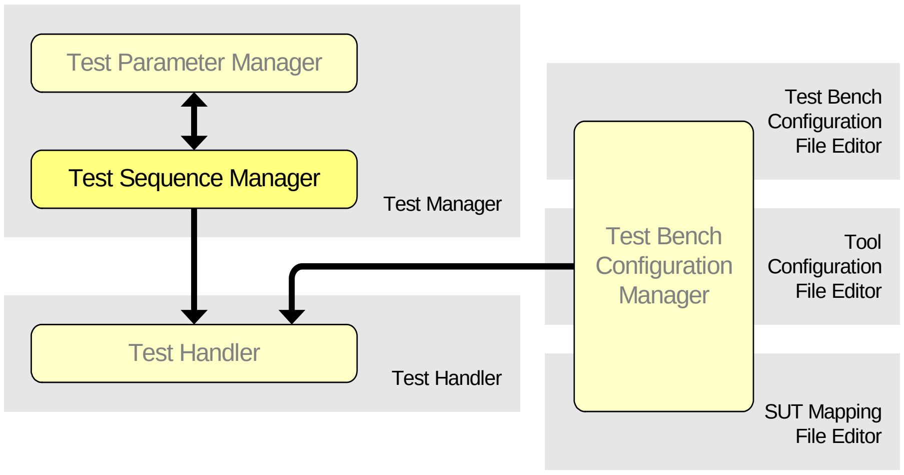
    The content of the dynamic arrays is merged.

  – Replace

    The content of the dynamic arrays is replaced completely.

4.3 The Tasks of the Test Sequence Manager

The Test Sequence Manager determines orders of execution and all other details which the Test Handler needs to execute a test for test cases of a UuT Group which have been assigned parameters.



The individual tasks of the Test Sequence Manager are:

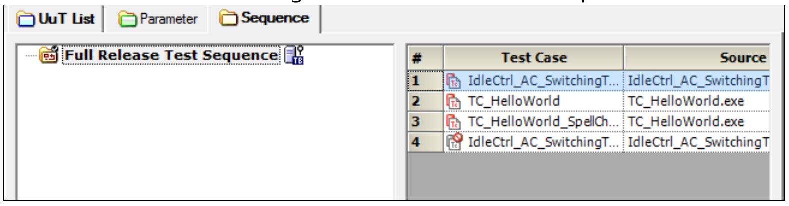
- creating and managing test sequences
- specifying the functionality for each test sequence
- specifying the order of execution in a test sequence
- defining Test Bench Initializations for test sequences

All these tasks are executed in the main workspace of Test Manager in the "Sequence" tab.

4.3.1 The "Sequence" Tab

In the "Sequence" tab (Fig. 4-13) all sequences of the UuT Group which were selected in the Project Explorer are displayed on the left.

The test cases belonging to the selected sequence and other information on these are shown on the right-hand side of the "Sequence" tab.



#	Test Case	Source
1	IdleCtrl_AC_SwitchingT...	IdleCtrl_AC_SwitchingT
2	TC_HelloWorld	TC_HelloWorld.exe
3	TC_HelloWorld_SpellCh...	TC_HelloWorld.exe
4	IdleCtrl_AC_SwitchingT...	IdleCtrl_AC_SwitchingT

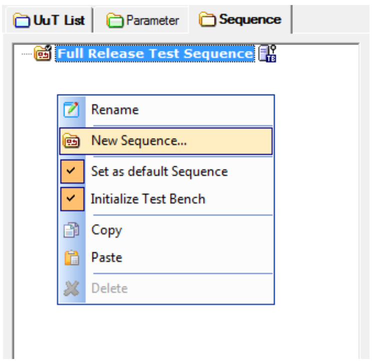
Fig. 4-13 The "Sequence" Tab

4.3.2 Working with Test Sequences

This section contains information on working with test sequences.

To create a test sequence

- Open the project to be edited.
- Select the desired UuT Group in the Project Explorer.
- Click the list of sequences in the "Sequence" tab using the right-hand mouse button.
- Select **New Sequence** from the shortcut menu.



The "(Sequence) Property Editor" opens.

- Enter a name for the new sequence and possibly a comment.
- Click **OK**.

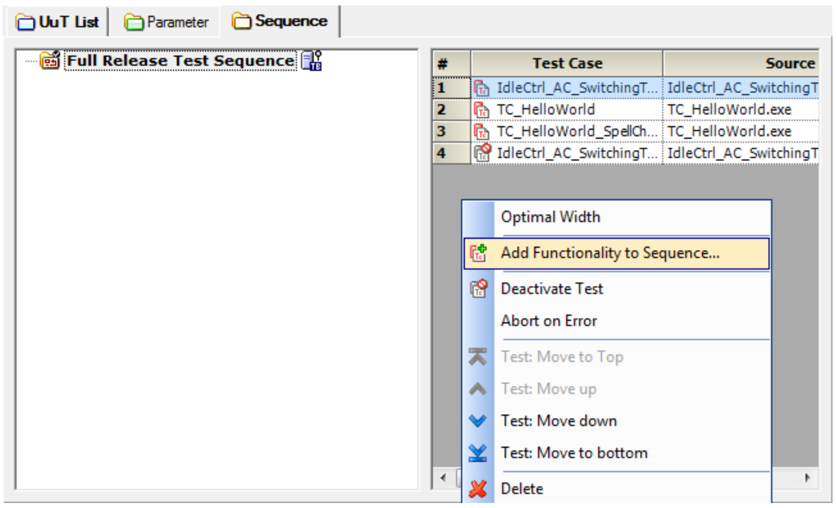
The new test sequence is created. As it is the only sequence so far, it is automatically a default sequence (shown by the use of bold).

To add a test case to the test sequence

If you want to add functionality to a test sequence, proceed as follows:

- Select a sequence from the "Sequence" tab.
- Right-click the right-hand side of the "Sequence" tab (i.e. the list of test cases).

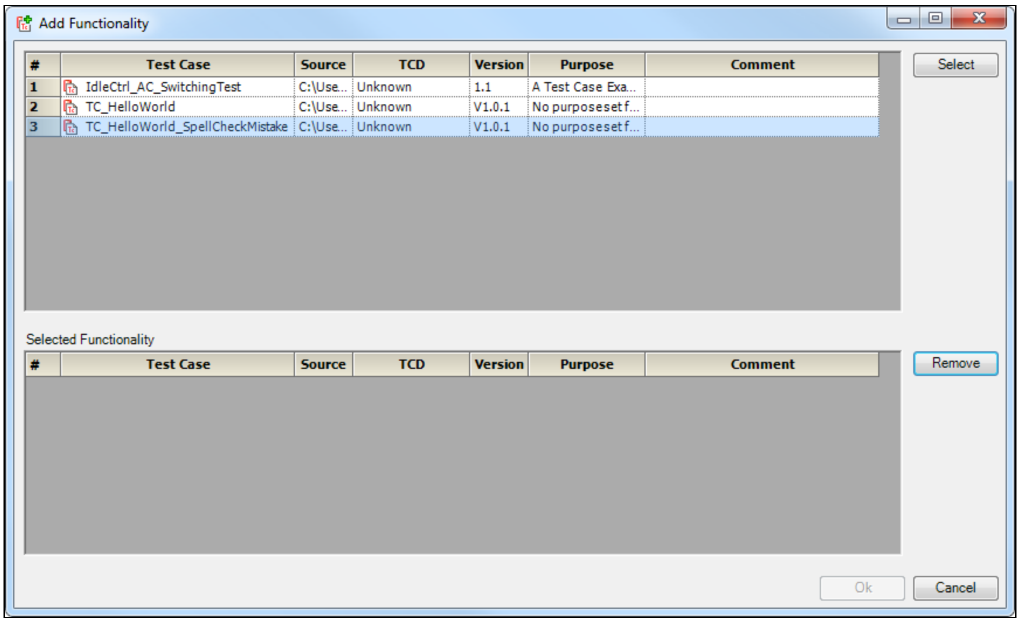
- Select **Add Functionality to Sequence** from the shortcut menu.



#	Test Case	Source
1	IdleCtrl_AC_SwitchingT...	IdleCtrl_AC_SwitchingT
2	TC_HelloWorld	TC_HelloWorld.exe
3	TC_HelloWorld_SpellCh...	TC_HelloWorld.exe
4	IdleCtrl_AC_SwitchingT...	IdleCtrl_AC_SwitchingT

The "Add Functionality" window opens. All test cases of the UuT Group are displayed in the upper part of the window.

- Select all test cases you want to add to the test sequence (multiple selection with <CTRL>).
- Click **Select**.



#	Test Case	Source	TCD	Version	Purpose	Comment
1	IdleCtrl_AC_SwitchingTest	C:\Use...	Unknown	1.1	A Test Case Exa...	
2	TC_HelloWorld	C:\Use...	Unknown	V1.0.1	No purpose set f...	
3	TC_HelloWorld_SpellCheckMistake	C:\Use...	Unknown	V1.0.1	No purpose set f...	

#	Test Case	Source	TCD	Version	Purpose	Comment

The selected test cases are added to the "Selected Functionality" list.

- If you want to remove one or more test cases from this list, select **Remove**.

- To add the selected test cases to the sequence, click **OK**.

Note

Test cases can also be inserted into the list of sequences from the Functionality Browser using drag-and-drop.

To change the order of the test cases within a test sequence

- Select the test case the position of which you want to change.
- Select **Test: Move** *nnn* from the shortcut menu.

  The test case is moved according to the selected menu item.

To rename a test sequence

- Select a sequence from the "Sequence" tab.
- Select **Rename** from the shortcut menu.
- Enter a new name.
- Click **OK**.

  The test sequence is renamed.

To make a test sequence the Default Sequence

To make a sequence the Default Sequence, proceed as follows:

- Select a sequence from the "Sequence" tab.
- Select **Set as Default Sequence** from the shortcut menu.

  The selected sequence is made the Default Sequence; this is indicated by the tick in the sequence icon and by the name of the sequence being shown in boldface.

To add Test Bench Initialization to a test sequence

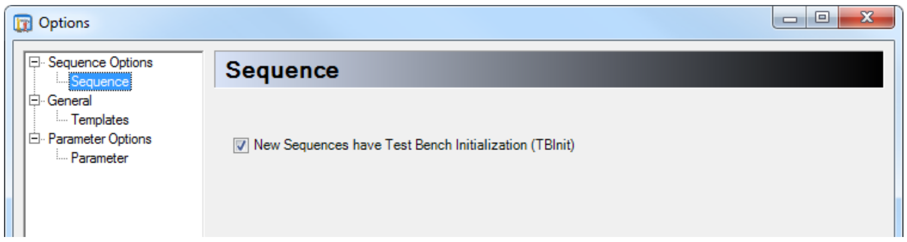
The Test Sequence Manager can integrate Test Bench Initialization/Finalization functionality into its test sequence. This creates a list of available ports and carries out the relevant initializations. To add a Test Bench Initialization to a sequence, proceed as follows:

- Select a sequence from the "Sequence" tab.
- Select **Initialize Test Bench** from the shortcut menu.

The Test Bench Initialization is added to the sequence. This is indicated by the symbol after the sequence name.

You can also have a Test Bench Initialization added automatically when a sequence is created.

- To do this, select **Tools** → **Options** and "Sequence Options/Sequence".
- Activate the option "New Sequences have Test Bench Initialization (TBInit)".



☑ New Sequences have Test Bench Initialization (TBInit)

- Click **OK**.

Each newly created sequence now automatically has a Test Bench Initialization.

To delete a test sequence

- Select a sequence from the "Sequence" tab.
- Select **Delete** from the shortcut menu.

The sequence is deleted.

Note

A default sequence cannot be deleted!

To deactivate/reactivate a test case

If you do not want certain test cases of a sequence to be executed, you can deactivate these (temporarily).

- Select the test case you want to deactivate.
- Select **Deactivate Test** from the shortcut menu.

The test case is deactivated which is also shown by the changed icon.

- To reactivate this test case, select **Reactivate Test**.

Abort on Error

If you want to abort an entire test sequence in the Test Handler if a test case ends with the verdict ”Inconc” or ”Error”, you can do this by setting the ”Abort on Error” flag. This means you can, for example, avoid a sequence being run completely if an error occurs on initialization.

- Select the test case of the sequence for which you want to set this flag.
- Select **Abort on Error** from the shortcut menu.

  The flag is set, indicated by the changed icon in front of the test case.

Note

The sequence continues to be run if the verdict of the selected test case is "Pass" or "Fail".

To remove test cases from a test sequence

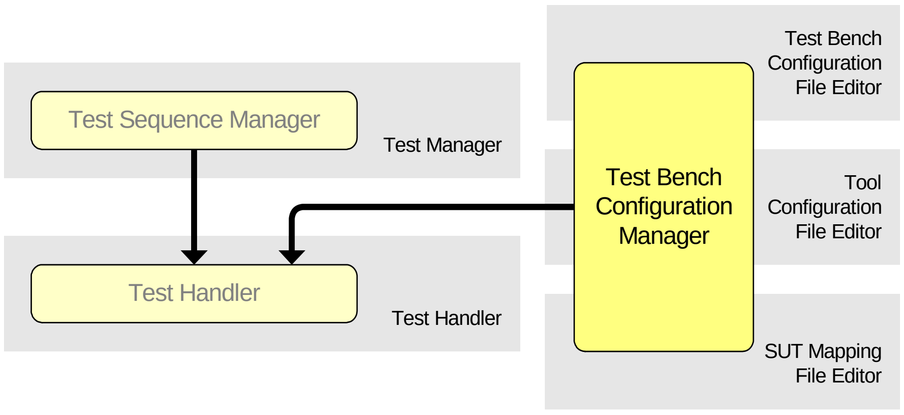
If you want to remove a test case from the sequence permanently, proceed as follows:

- Select the test case you want to delete.
- Select **Delete** from the shortcut menu.

  The test case is deleted from the sequence.

4.4 The Tasks of the Test Bench Configuration Manager

The Test Bench Configuration Manager (TBCM) provides all tools to make a specific test project executable.


Test Sequence Manager
Test Manager
Test Handler
Test Handler
Test Bench Configuration Manager
Test Bench Configuration File Editor
Tool Configuration File Editor
SUT Mapping File Editor

The relevant editors of the configuration files are the workspace of the Test Bench Configuration Manager.

In this workspace, the Test Bench Configuration Manager manages the following files:

- Test Bench Configuration File

  The main purpose of the Test Bench Configuration File (TBCF) is to map port instances of the test cases to specific tool instances.

- Tool Configuration File

  A Tool Configuration File (TCF) contains a valid configuration of all tools of the test bench which are required to run the test project.

- SUT Mapping File

  This file contains the mapping of the test-bench-independent logical test label to the labels which are used in the current test environment.

For a detailed description of these files and the relevant editors, refer to the chapter "Configuration Files and their Editors" on page 147.

From Version 3.3 of LABCAR-AUTOMATION you can configure the test bench using the Configuration Wizard described below.

4.4.1 The Configuration Wizard

The Configuration Wizard is a tool that facilitates your start to developing test cases and creating test projects and test bench configurations. It acts as an instructor and enables you to create a test bench configuration and a suitable LABCAR-AUTOMATION project with just a few clicks of the mouse.

There are two version of the Configuration Wizard: “Standard” and “Professional”. In the “Standard” version, most entries are filled with fixed values – the “Professional” version makes it possible to change these values.

To understand the individual steps and the significance of the configuration files, refer to the detailed information in "Working with LABCAR-AUTOMATION 4.2.3" on page 51 and "Configuration Files and their Editors" on page 147.

Overview

The Configuration Wizard is used to generate a test bench configuration (*.tbc) with the required tool configurations (*.tcf). If not already available, an SUT Mapping File (*.smf) is generated.

In the test bench configuration, tools (and thus implicitly tool configuration files) are assigned to the selected ports. The SUT Mapping File contains the mapping of the (tool-independent) test labels, as used in the test case, to the tool labels of the tools used in each case (see Fig. 1-5 on page 24).

In addition, an (“empty”) LABCAR-AUTOMATION project with

- Default UuT
- Default test case (from the “Default” C# project)
- Default parameterization
- Default sequence

is created and everything is saved in a shared folder structure.

The “Default” C# project is used as a template and already has the port configurations integrated in the “default test case”.

Tool-specific data, such as LABCAR-OPERATOR projects, experiment and workspace for LABCAR-EE or INCA databases, is not generated.

Creating Projects with the Configuration Wizard

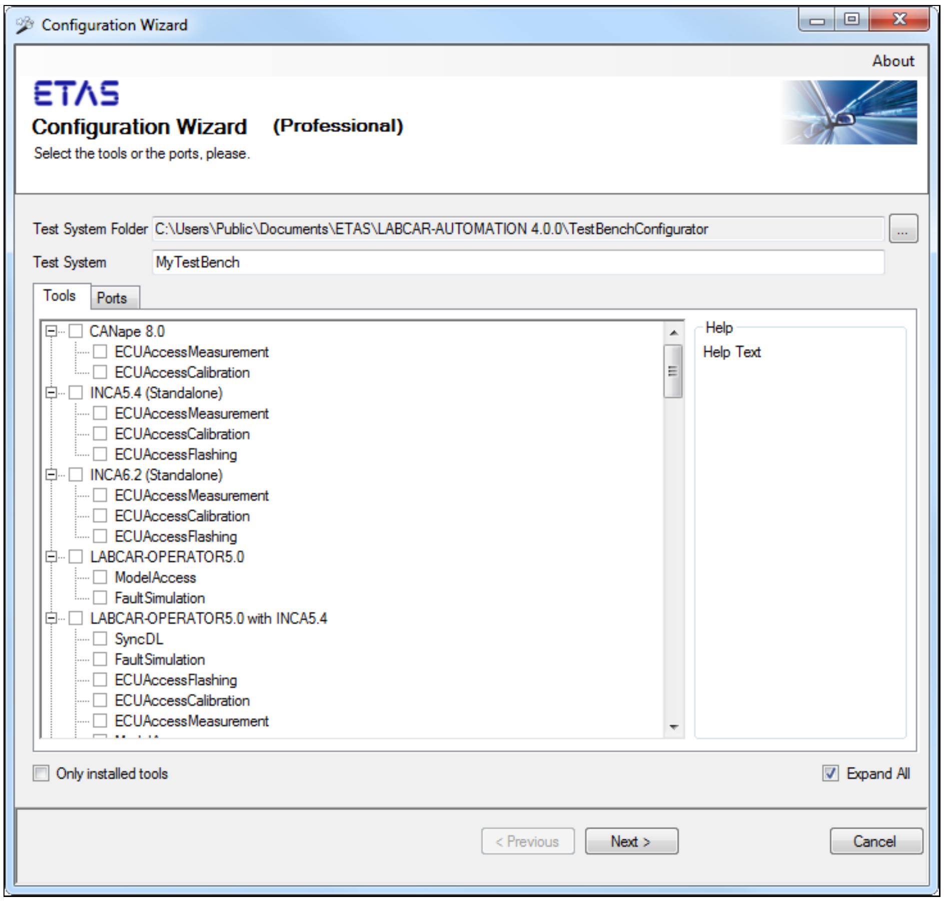
Note

The following descriptions refer to the "Professional" version of the Configuration Wizard – path changes are not possible in the "Standard" version!

To open the Configuration Wizard

- Select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2** → **Configuration Wizard**.

  The Configuration Wizard opens.



Configuration Wizard

About

ETAS
Configuration Wizard (Professional)
Select the tools or the ports, please.

Test System Folder C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\TestBenchConfigurator

Test System MyTestBench

Tools | Ports

- CANape 8.0
  - ECUAccessMeasurement
  - ECUAccessCalibration
- INCA5.4 (Standalone)
  - ECUAccessMeasurement
  - ECUAccessCalibration
  - ECUAccessFlashing
- INCA6.2 (Standalone)
  - ECUAccessMeasurement
  - ECUAccessCalibration
  - ECUAccessFlashing
- LABCAR-OPERATOR5.0
  - ModelAccess
  - FaultSimulation
- LABCAR-OPERATOR5.0 with INCA5.4
  - SyncDL
  - FaultSimulation
  - ECUAccessFlashing
  - ECUAccessCalibration
  - ECUAccessMeasurement

Help
Help Text

☐ Only installed tools
☑ Expand All

To select a directory for project data

- The current directory, in which data generated by the Configuration Wizard is stored, is shown in the "Test System Folder" box.
- Click the ... icon to select another directory.

To select a name for a project folder

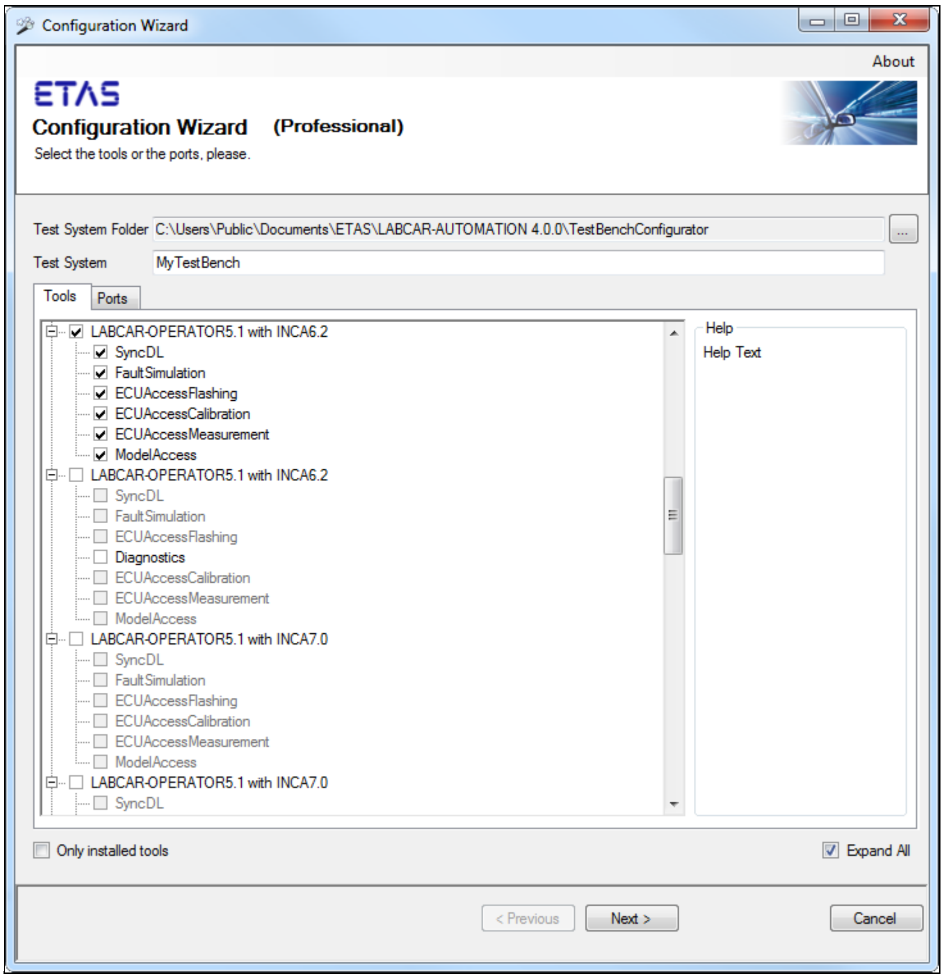
- Select a name for the folder in which (in the main directory selected above) the configuration to be created is to be saved.

Note

If a directory of this name already exists, the name is shown in red!

To select tools (or ports)

- Select one or more tools you want to use during the project from the "Tools" tab.



Select the tools or the ports, please.

Test System Folder C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\TestBenchConfigurator

Test System MyTestBench

Tools | Ports

- ✓ LABCAR-OPERATOR5.1 with INCA6.2
  - ✓ SyncDL
  - ✓ FaultSimulation
  - ✓ ECUAccessFlashing
  - ✓ ECUAccessCalibration
  - ✓ ECUAccessMeasurement
  - ✓ ModelAccess
- ☐ LABCAR-OPERATOR5.1 with INCA6.2
  - ☐ SyncDL
  - ☐ FaultSimulation
  - ☐ ECUAccessFlashing
  - ☐ Diagnostics
  - ☐ ECUAccessCalibration
  - ☐ ECUAccessMeasurement
  - ☐ ModelAccess
- ☐ LABCAR-OPERATOR5.1 with INCA7.0
  - ☐ SyncDL
  - ☐ FaultSimulation
  - ☐ ECUAccessFlashing
  - ☐ ECUAccessCalibration
  - ☐ ECUAccessMeasurement
  - ☐ ModelAccess
- ☐ LABCAR-OPERATOR5.1 with INCA7.0
  - ☐ SyncDL

Help

Help Text

☐ Only installed tools

✓ Expand All

The relevant ports are shown under the selected tool entry ("Expand All" option). All ports affected by this selection (even if these are also to be found with other tools) are shown grayed out.

Note

Select the "Only installed tools" option to exclusively display the tools available on your system.

In the above example, the "LABCAR-OPERATOR V5.0 with INCA 6.2" tools were selected.

- Toggle to the "Ports" tab.

  The (available) ports are displayed and, underneath, the tools required by these ports.

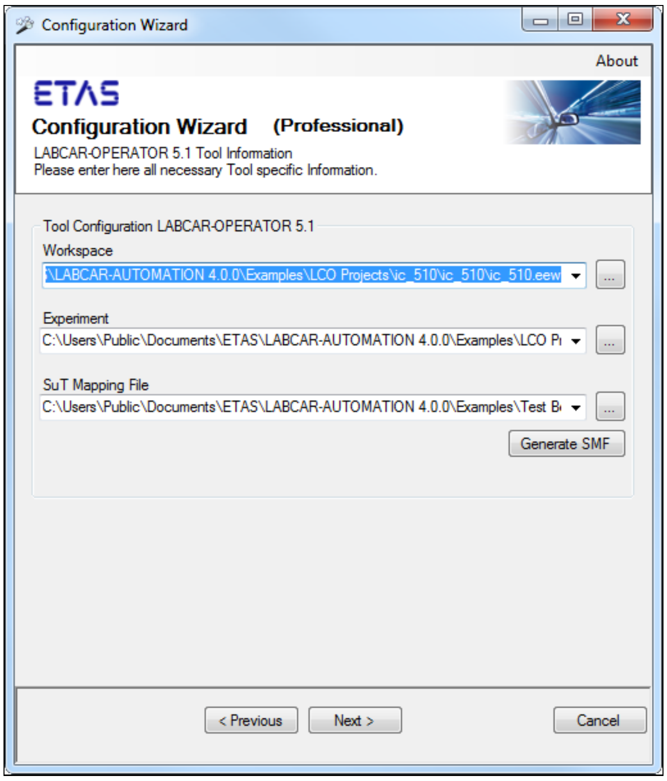
Note

Before the next step, ensure that the "Test Handler" application and the LABCAR-AUTOMATION Engine Controller are closed.

- Click **Next**.

To enter tool-specific information (LABCAR-OPERATOR)

In the next window, you will enter further tool-specific information (tool configuration).



Configuration Wizard

ETAS Configuration Wizard (Professional)

LABCAR-OPERATOR 5.1 Tool Information

Please enter here all necessary Tool specific Information.

Tool Configuration LABCAR-OPERATOR 5.1

Workspace: \LABCAR-AUTOMATION 4.0.0\Examples\LCO Projects\ic_510\ic_510\ic_510.eew

Experiment: C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\LCO Pr

SuT Mapping File: C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Be

- The following information is required for the "LABCAR-OPERATOR" tool:
  - Workspace

    The file (*.eew) that contains the experiment environment for the LABCAR-OPERATOR experiment.
  - Experiment

    The LABCAR-OPERATOR experiment (*.eex)
  - An SUT Mapping File

To create an SUT Mapping File for a LABCAR-OPERATOR experiment

If no SUT Mapping File is available, you can create a "template" for this purpose.

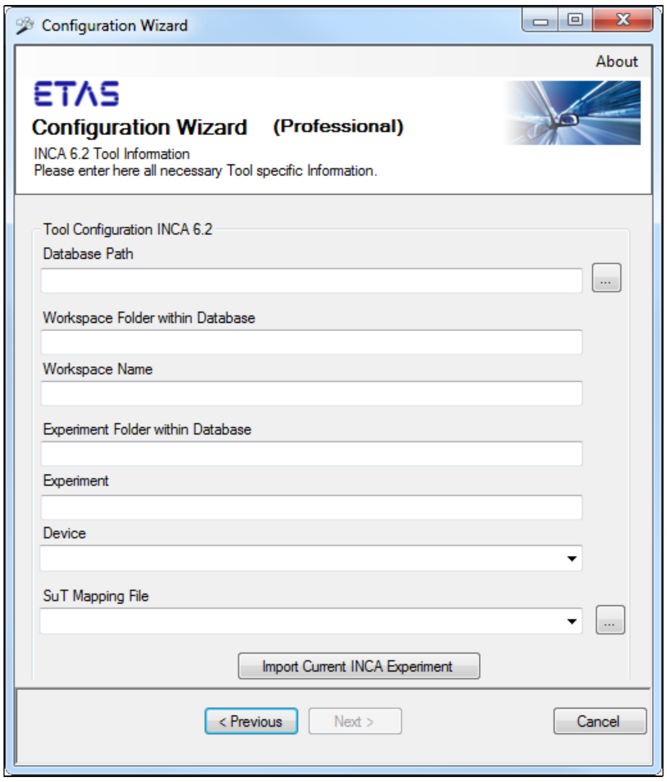
- Select **Generate SMF**.

  The experiment opens in the experiment environment and all tool labels of the experiment are written to a new SUT Mapping File.

  The test labels have the same name as the tool labels and can be edited later with the LABCAR-SMFEditor (see the online help of the SMFE).
- Click **Next**.

To enter tool-specific information (INCA)

If – as selected in the example above – you also want to configure measure and calibration access with INCA, further information is required:


Configuration Wizard
About
ETAS
Configuration Wizard (Professional)

INCA 6.2 Tool Information
Please enter here all necessary Tool specific Information.

Tool Configuration INCA 6.2
Database Path
Workspace Folder within Database
Workspace Name
Experiment Folder within Database
Experiment
Device
SuT Mapping File

...
...
Import Current INCA Experiment
< Previous
Next >
Cancel

- Enter the necessary data (see the description of the tool configuration for INCA on page 170) manually:
  - Database Path: path to the INCA database
  - Workspace Folder within Database
  - Workspace Name
  - Experiment Folder within Database
  - Experiment
  - Device

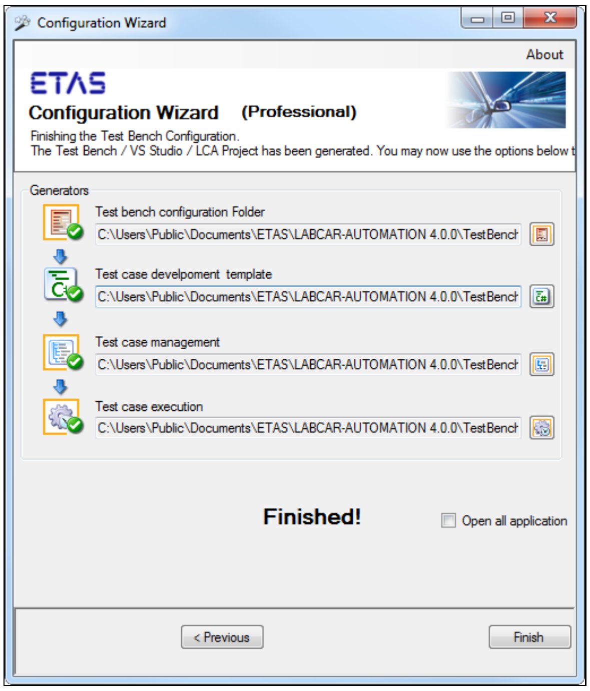
  – SuT Mapping File

or (from INCA V6.2.1)

- Open the experiment in INCA.
- Click **Import Current INCA Experiment**.

  The experiment data is accepted in the relevant boxes (see above).
- Click **Next**.

  Once the last information has been entered, the project files are generated.


Configuration Wizard

About

Configuration Wizard (Professional)

Finishing the Test Bench Configuration.
The Test Bench / VS Studio / LCA Project has been generated. You may now use the options below t

Generators

Test bench configuration Folder
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\TestBench

Test case develpoment template
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\TestBench

Test case management
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\TestBench

Test case execution
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\TestBench

Finished!

☐ Open all application

< Previous   Finish

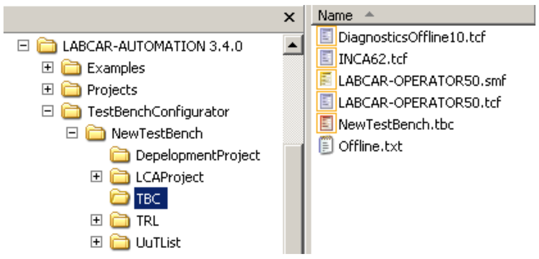
The Generated Files and Directories

These are:

- Test bench configuration folder

  The files for the test bench configuration (*.tcf, *.tbc, *.smf, ...)

  Clicking the icon opens the relevant folder.



LABCAR-AUTOMATION 3.4.0
- Examples
- Projects
- TestBenchConfigurator
  - NewTestBench
    - DepelopmentProject
    - LCAProject
    - TBC
    - TRL
    - UuTList

Name
- DiagnosticsOffline10.tcf
- INCA62.tcf
- LABCAR-OPERATOR50.smf
- LABCAR-OPERATOR50.tcf
- NewTestBench.tbc
- Offline.txt

- Test case development template

  A C# template for a test case:
  `C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2\`
  `TestBenchConfigurator\`*Project Name*`\`
  `DevelopmentProject\`*Project Name*`.csproj`

  Clicking the icon opens your development environment

- Test case management

  The LABCAR-AUTOMATION project and further files in the directory

  `C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2\`
  `TestBenchConfigurator\`*Project Name*`\LCAProject\`

  Clicking the icon opens the Test Manager for further project processing.

Note

The UuT "Default" must be selected explicitly in the Project Explorer to ensure the UuT settings and test sequences are displayed!

- Test case execution

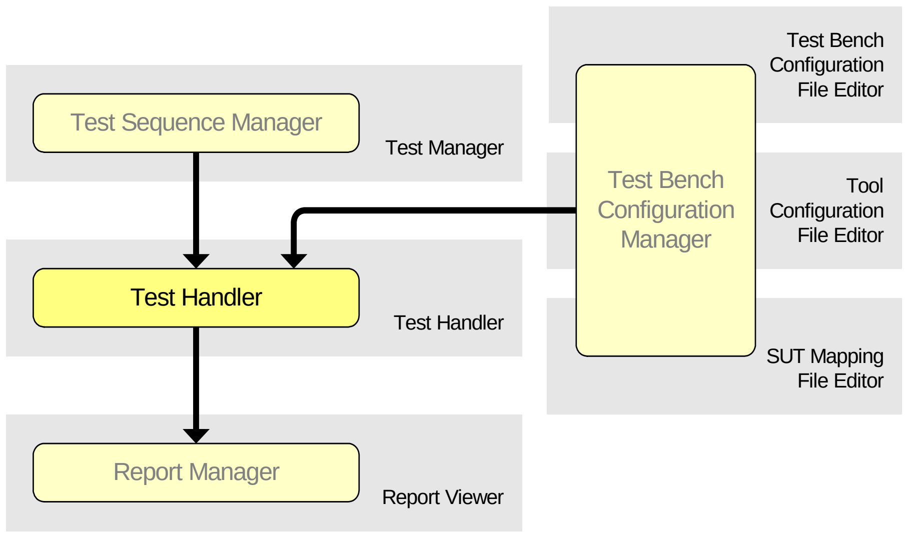
  Clicking the icon opens the Test Handler for running the project.

Note

The test run configuration (Test Project → Default → Test List → Default Test Run Configuration) must be selected explicitly for the test run to be displayed!

4.5 The Tasks of the Test Handler

The Test Handler's job is to execute the test. This means that it selects a UuT and thus implicitly a list of test sequences. Together with a valid Test Bench Configuration, it is then capable of carrying out test sequences. Test reports are created automatically once the sequences have been executed.



The Test Handler has the following tasks:

- selecting a test project, a UuT from this project and a test sequence for this UuT (see "Selecting a Test Sequence" on page 109)
- defining Test Run Configurations: activating or deactivating test cases within a test sequence (see "Editing Test Sequences" on page 111 and "Test Run Configurations" on page 113)
- selecting a Test Bench Configuration (see "Test Bench Configuration" on page 115)
- starting test execution (see "Test Execution" on page 118)
- management of the reports on the executed sequences including Test Bench Configuration, test results, project information etc. (see "Test Reports" on page 125)
- creating batch projects ("Working with Batch Projects" on page 126)

To start the user interface for the Test Handler

- From the Windows Start menu select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2** → **Test Handler**.
- The user interface for the Test Handler opens.

4.5.1 The Test Handler User Interface

Fig. 4-14 shows the Test Handler user interface without a test project being loaded.

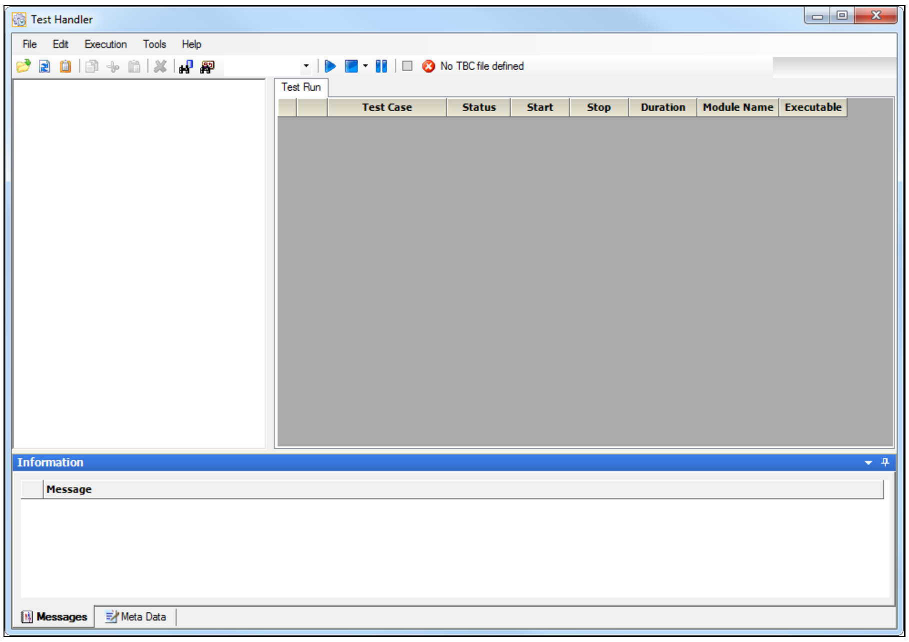
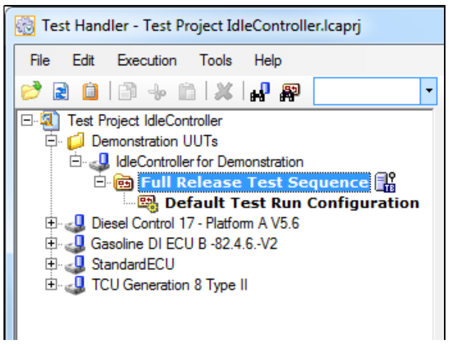


Fig. 4-14 The Test Handler User Interface

The individual areas of the user interface are described below.

The Project Explorer

The Project Explorer contains the structure of the test project (see "To open a project" on page 110) with the UuTs and the test sequences assigned to them – the desired test sequence can be selected here. Below the test sequence are what are referred to as Test Run Configurations (see "Test Run Configurations" on page 113) created by the Test Handler and below these, in turn, the relevant test reports (see "Test Reports" on page 125).

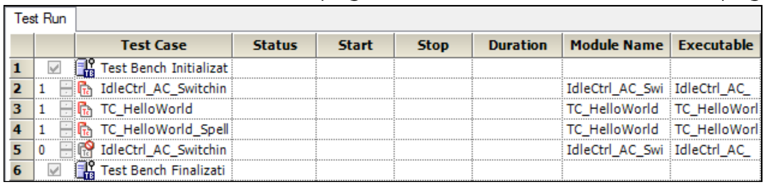

Test Handler - Test Project IdleController.lcaprj
File Edit Execution Tools Help

- Test Project IdleController
  - Demonstration UUTs
    - IdleController for Demonstration
      - Full Release Test Sequence
        - Default Test Run Configuration
  - Diesel Control 17 - Platform A V5.6
  - Gasoline DI ECU B -82.4.6.-V2
  - StandardECU
  - TCU Generation 8 Type II

The Tabular View

This table shows the test cases of the test sequence selected in the Project Explorer. The test cases are shown in the order of their execution – defined by the Test Sequence Manager.

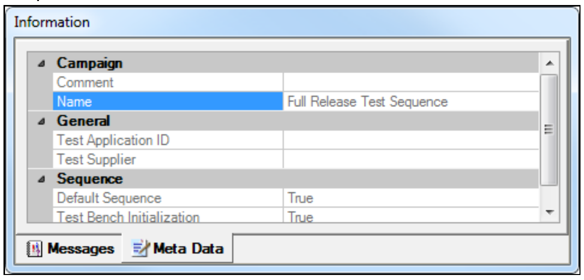
The view is particularly used to determine repetitions or to deactivate test cases as well as the status display during test execution (see "Iteration Number: Active and Inactive Test Cases" on page 112 and "Test Execution" on page 118).



			Test Case	Status	Start	Stop	Duration	Module Name	Executable
1		✓	Test Bench Initializat						
2	1		IdleCtrl_AC_Switchin					IdleCtrl_AC_Swi	IdleCtrl_AC_
3	1		TC_HelloWorld					TC_HelloWorld	TC_HelloWorl
4	1		TC_HelloWorld_Spell					TC_HelloWorld	TC_HelloWorl
5	0		IdleCtrl_AC_Switchin					IdleCtrl_AC_Swi	IdleCtrl_AC_
6		✓	Test Bench Finalizati						

The "Information" Window

The "Information" window has two tabs: One for messages ("Messages") and one for meta data ("Meta Data") on the object currently selected (e.g. a test sequence or a UuT).



Campaign	
Comment	
Name	Full Release Test Sequence
General	
Test Application ID	
Test Supplier	
Sequence	
Default Sequence	True
Test Bench Initialization	True

Fig. 4-15 "Meta Data" of a Test Sequence

4.5.2 Selecting a Test Sequence

There are several steps to selecting a test sequence:

1. selecting a project
2. selecting a UuT from the project
3. selecting a sequence of the UuT

To open a project

To open a project proceed as follows:

- Select **File** → **Open Project**.

or

- Click the **Open Project** icon.

  A file selector window opens.
- Select the project to be opened.

  If the project to be opened was used recently, it will be in the list of projects opened recently.
- Select **File** → **Recent Projects** → *project_name*.

  The project opens and is shown in the "UuT" in its structure (UuTs, test sequences and Test Run Configurations).

Note

If you do not know the name of the project to be opened, you can search directories for project attributes. Proceed as follows:

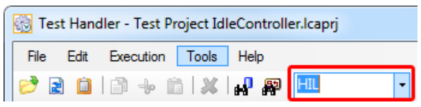
Selecting UuT and Test Sequence

If the open project is not too extensive, the desired UuT folder can be opened by clicking the mouse and one of the relevant test sequences selected.

If, however, the list of UuTs and test sequences is very extensive, the search function should be used:

To search a UuT

- To find a UuT, enter the character string you want to look for in the field in the toolbar.



- In the main menu, select **Edit** → **Find UuT**.

or

- Click the **Find UuT** icon in the toolbar.

  If a UuT is found, whose name corresponds to the search criteria, it is selected in the Project Explorer.

  If several UuTs are found which correspond to the search, they are selected consecutively by repeatedly clicking **Find UuT**.

To search for a test sequence

- To find a test sequence, enter the character string you want to look for in the field in the toolbar.
- In the main menu, select **Edit** → **Find Sequence**.

or

- Click the **Find Sequence** icon in the toolbar.

  If a test sequence is found, whose name corresponds to the search criteria, it is selected in the Project Explorer.

  If several test sequences are found which correspond to the search, they are selected consecutively by repeatedly clicking **Find Sequence**.

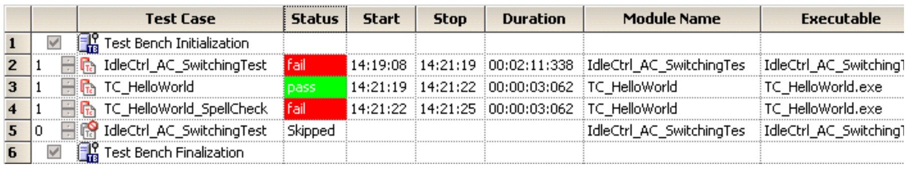
4.5.3 Editing Test Sequences

Before executing a Test Run Configuration (see "Test Run Configurations" on page 113), it may be necessary (e.g. when searching for errors) to deactivate partial functionality or specify that certain test cases are executed several times.

The Tabular View of Test Handler

In this table, the test cases belonging to the particular test sequence are displayed in the order of their execution. If test cases belonging to the test sequence have already been deactivated by the Test Sequence Manager, these are no longer included in the list – i.e. the table only contains executable test cases.

The following figure shows the tabular view after a test sequence has been executed.



		Test Case	Status	Start	Stop	Duration	Module Name	Executable
1	✓	Test Bench Initialization						
2	1	IdleCtrl_AC_SwitchingTest	fail	14:19:08	14:21:19	00:02:11:338	IdleCtrl_AC_SwitchingTes	IdleCtrl_AC_Switching1
3	1	TC_HelloWorld	pass	14:21:19	14:21:22	00:00:03:062	TC_HelloWorld	TC_HelloWorld.exe
4	1	TC_HelloWorld_SpellCheck	fail	14:21:22	14:21:25	00:00:03:062	TC_HelloWorld	TC_HelloWorld.exe
5	0	IdleCtrl_AC_SwitchingTest	Skipped				IdleCtrl_AC_SwitchingTes	IdleCtrl_AC_Switching1
6	✓	Test Bench Finalization						

Fig. 4-16 Display of a Test Sequence in Test Handler

The individual columns have the following meaning:

- 1st Column

  Line number
- Iteration number

  This is where you can set how many times the test case is to be executed (see "Iteration Number: Active and Inactive Test Cases" on page 112).
- Test Case

  Instance name of the test case.
- Status *

  Result ("verdict") after execution of the test case – possible values are:

  – none

  – pass

  - fail
  - inconc
  - error
  - interrupted
  - skipped
- Start *

  Start of execution.
- Stop *

  End of execution.
- Duration *

  Duration
- Module Name

  Name of the TTCN-Module of the test case.
- Executable

  Path and executable of the test case.

* The content of the columns "Status", "Start", "Stop" and "Duration" is filled while the test sequence is run and is not visible until a sequence has been completed as long as the project is not closed.

Iteration Number: Active and Inactive Test Cases

At the start, all test cases have the iteration number 1 – this means that they are part of the test sequence. Basically an iteration number greater than or equal to 1 means that this is an activated test case.

A test case is deactivated by the iteration number being set to zero.

Note

No changes can be made to the Default Test Run Configuration!

To change the iteration number

There are various ways of changing the iteration number:

- Enter a number directly

or

- Use the buttons with the arrows.

or

- Select a test case and press the right-hand mouse button.

  The shortcut menu opens.
- Select **Deactivate** to set the iteration number to zero.
- Select **Set to Default** to set the iteration number to 1.

4.5.4 Test Run Configurations

A Test Run Configuration consists of a sequence of test cases assigned iteration numbers. When a test project is opened for the first time in the Test Handler, a Default Run Configuration is created for every test sequence; it contains every test case with the iteration number = 1.

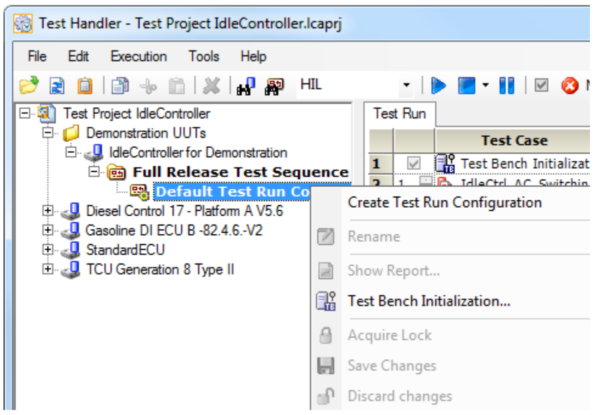
As many further Test Run Configurations (derived from this Default Test Run Configuration) as you like can be added.

To add a Test Run Configuration

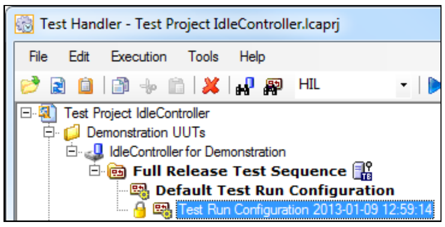
- Select **Edit** → **Add Test Run Configuration** from the main menu of the Test Handler.

or

- Right-click in the project window and select **Create Test Run Configuration**.


Test Handler - Test Project IdleController.lcaprj
File
Edit
Execution
Tools
Help
HIL
Test Project IdleController
Demonstration UUTs
IdleController for Demonstration
Full Release Test Sequence
Default Test Run Co
Diesel Control 17 - Platform A V5.6
Gasoline DI ECU B -82.4.6.-V2
StandardECU
TCU Generation 8 Type II
Test Run
Test Case
Test Bench Initializat
IdleCtrl_AC_Switchin
Create Test Run Configuration
Rename
Show Report...
Test Bench Initialization...
Acquire Lock
Save Changes
Discard changes

The Test Run Configuration is added. The padlock icon to the left of the name means that this Test Run Configuration is currently locked, i.e. it can be edited by the Test Handler.


Test Handler - Test Project IdleController.lcaprj
File
Edit
Execution
Tools
Help
HIL
Test Project IdleController
Demonstration UUTs
IdleController for Demonstration
Full Release Test Sequence
Default Test Run Configuration
Test Run Configuration 2013-01-09 12:59:14

- Change the iteration numbers and Test Bench Initialization/Finalization to suit your requirements.
- Select **Save Changes** from the shortcut menu to save your changes to the relevant sequence.
- Select **Discard Changes** to discard your changes.

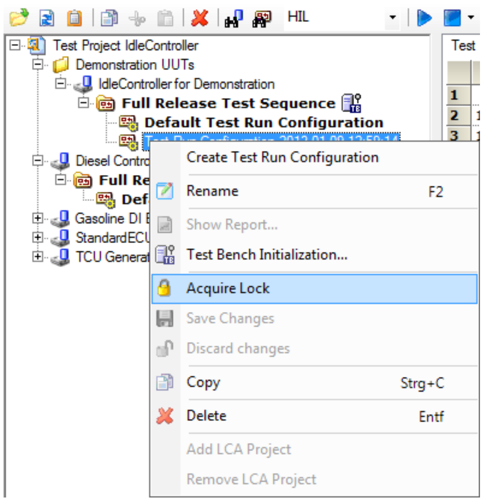
The changes made are accepted (or discarded respectively) and the lock canceled.

If the Test Run Configuration is selected with the mouse, it is shown in the tabular view – this means that the Test Handler can save several Test Run Configurations for one test sequence and simply toggle between them.

To lock a Test Run Configuration

If you want to make changes to a Test Run Configuration (e.g. change the iteration number of a test case), you must first lock it for your own individual access.

- Select the Test Run Configuration in the project window.
- Select **Acquire Lock** from the shortcut menu.

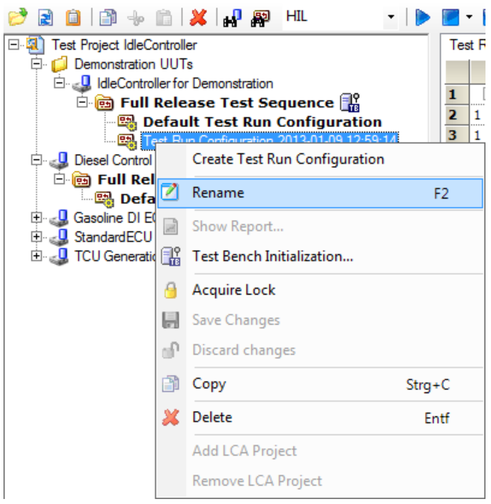


The Test Run Configuration is locked. This is indicated by the padlock icon to the right of the name of the Test Run Configuration.

- To cancel the lock again, select **Save Changes** or **Discard Changes** from the shortcut menu.

To rename or delete Test Run Configurations

- To rename a Test Run Configuration, select the relevant Test Run Configuration.
- Right-click it and select **Rename**.



Now you can change the name.

- To delete the Test Run Configuration, select **Delete**.

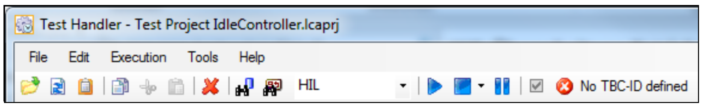
Note

If a Test Run Configuration is deleted, all the reports belonging to it are also deleted!

4.5.5 Test Bench Configuration

Before test sequences are executed, the Test Handler has to define a Test Bench Configuration – this takes place by selecting a Test Bench Configuration File.

If no Test Bench Configuration File has been defined, this is indicated in the toolbar.



To select a Test Bench Configuration

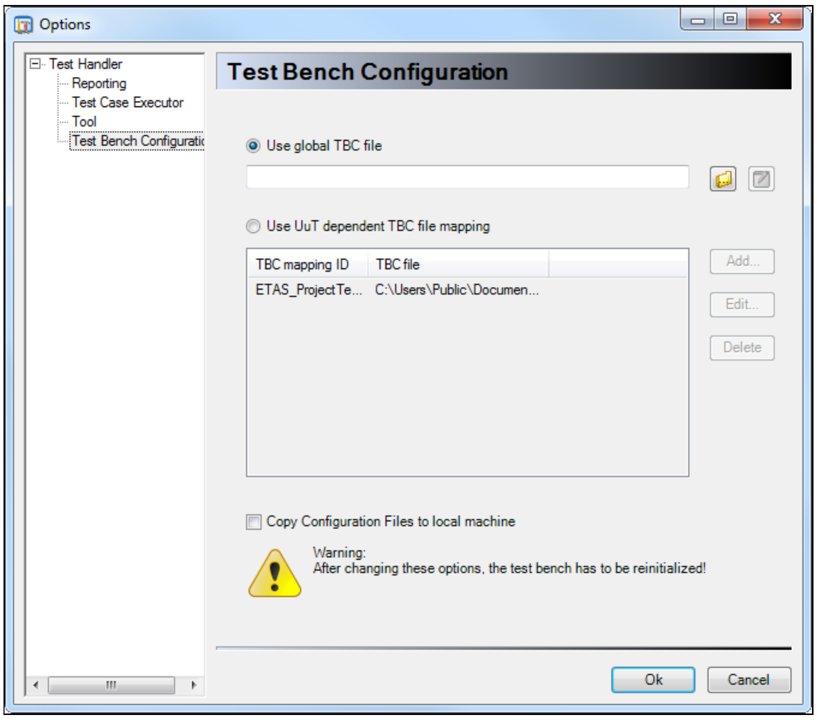
- Click the **Modify TBC options** icon in the toolbar.

or

- Select **Tools** → **Options**.

  The "Options" window opens.
- Select "Test Bench Configuration" in the left-hand part of the window.

  The Test Bench Configuration settings are shown.



Test Bench Configuration

(● Use global TBC file)

○ Use UuT dependent TBC file mapping

TBC mapping ID	TBC file
ETAS_ProjectTe...	C:\Users\Public\Documen...

☐ Copy Configuration Files to local machine

Warning:
After changing these options, the test bench has to be reinitialized!

- If you want to use a global Test Bench Configuration File, select the option "Use global TBC File".
- Click the folder symbol to select a Test Bench Configuration File (*.tbc).

  A file selector window opens.
- Select the required Test Bench Configuration File and click **OK**.
- To edit a selected file, click the **Edit the TBC file** icon.

  The Test Bench Configuration File Editor is launched and the selected file loaded.
- The "Use UuT dependent TBC file mapping" option enables the Test Bench Configuration to be selected according to the UuT (shown by an ID).
- If you select the option "Copy Configuration Files to local machine", local copies of configuration files (TBCF, SMF, TCF) on a network server are made.
- If you do not want to set any other options, quit the window with **OK**.

Note

Once a Test Bench Configuration File has been selected, this selection is saved in the registry and is thus available for all Test Handler sessions. This selection can, however, be changed any time.

Before a test execution is started, there is an automatic check to see whether the selected Test Bench Configuration is consistent with the settings of the current test sequence.

This check can also be carried out manually by the user.

To check consistency

- Select **Tools** → **Check Consistency**

or

- Click the **Check Consistency** icon in the toolbar.

  The check is carried out and the result is displayed in the "Information" window in the "Messages" tab.

Test Bench Overview

For an overview of the Test Bench Configuration, select **Tools** → **Test Bench Overview** in the Test Handler menu.

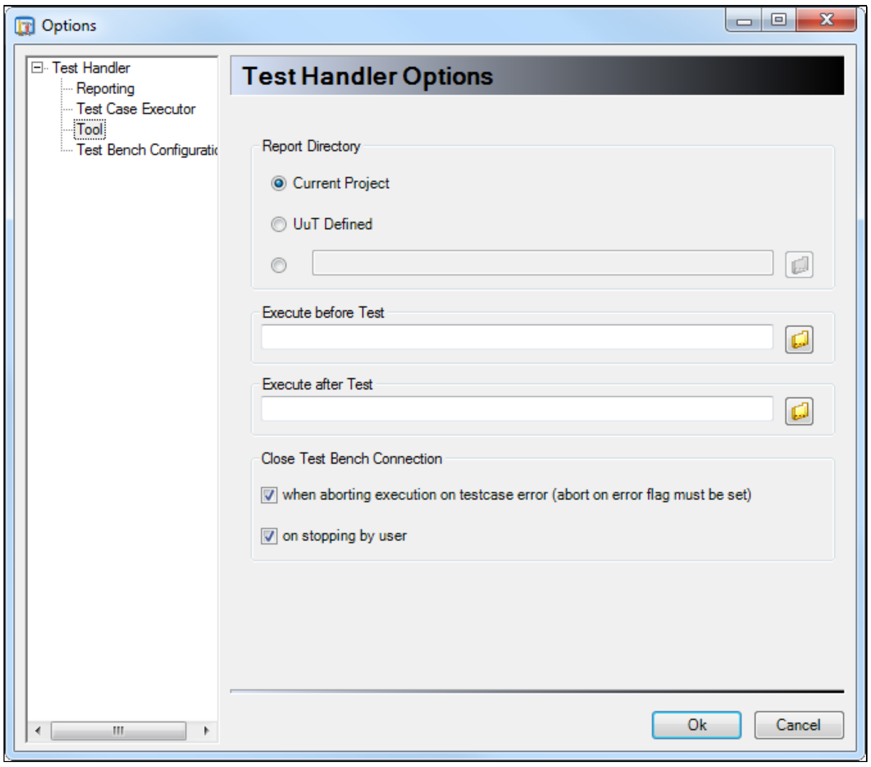
4.5.6 Specifying the Report Directory

To specify the directory in which the report files (see "Test Reports" on page 125) are to be stored, proceed as follows:

- Select **Tools** → **Options**.

  The "Options" window opens.
- Select "Tool" on the left-hand side of the window.

  The Test Handler Options are displayed.



Test Handler Options

Report Directory

[✓ Current Project] UuT Defined

Execute before Test

Execute after Test

Close Test Bench Connection

☑ when aborting execution on testcase error (abort on error flag must be set)

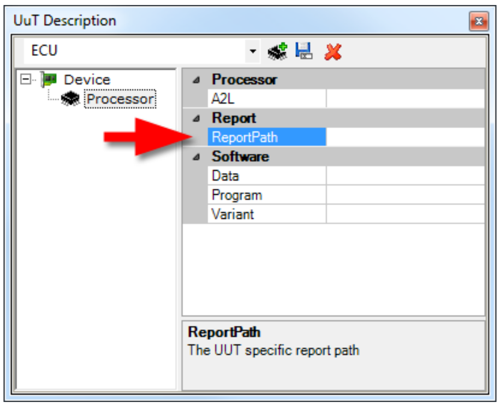
☑ on stopping by user

Under "Report Directory" you can select one of the following options:

- Current Project

  The data is stored in the current project directory.
- UuT Defined

  The path for the report directory is a property of the UuT and is specified in the UuT Description (see "Processing and Managing UuT Descriptions/ Environments" on page 192).



ReportPath
The UUT specific report path

- Any path

  Click the folder icon to select (or alternatively enter manually) the directory in which the test report should be stored.

4.5.7 Test Execution

After a Test Bench Configuration has been selected, the Test Run Configuration can be executed.

Before Test Execution

Before a test is run, it might be useful to trigger preparatory measures (outside LABCAR-AUTOMATION 4.2.3). This can be achieved by the Test Handler specifying an executable file. The same applies for the execution of a file after test execution (see "After Test Execution" on page 119).

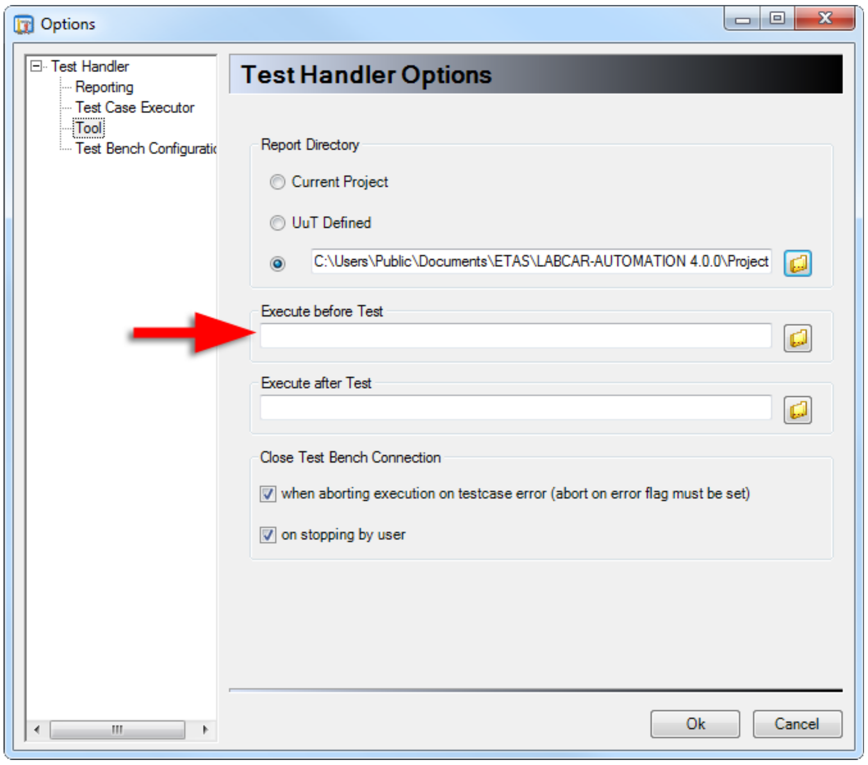
To specify the application

- Select **Tools** → **Options** from the Test Handler main menu.

  The "Options" window opens.
- Enter the application to be run in the "Execute before Test" field.

or

- Click the folder icon and select the required file.



Test Handler Options

Report Directory
- Current Project
- UuT Defined
- [✓] C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Project

Execute before Test

Execute after Test

Close Test Bench Connection
- [✓] when aborting execution on testcase error (abort on error flag must be set)
- [✓] on stopping by user

- Click **OK**.

  The application selected is launched before the test is executed.

After Test Execution

Once a test has been run, further actions should be initiated automatically (outside LABCAR-AUTOMATION 4.2.3). These might, for example, be:

- Copying the test report into specific directories for further processing
- Sending an e-mail to the person executing the test

Once a test has been completed, an application can be launched which is run in the Windows Command Prompt. The following parameters are passed (in this order):

1. The verdict (string)
2. The UuT ID (string)
3. The "test-sequence-modified" flag (Boolean)

   Is set if the "Number of Iterations" has changed in a test case in the Test Run Configuration.
4. The directory name of the test report (string)
5. The "test-sequence-completed" flag (Boolean)

   Is set if the test sequence has been completed.

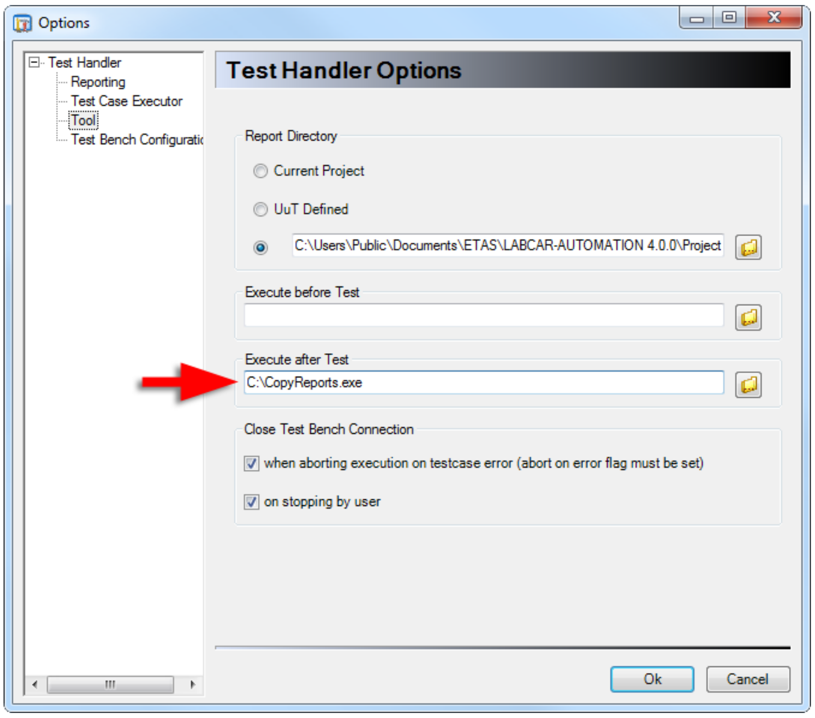
To specify the application

- Select **Tools** → **Options** from the Test Handler main menu.

  The "Options" window opens.
- Enter the application to be run in the "Execute after Test" field.

or

- Click the folder icon and select the required file.


Options

Test Handler Options

Report Directory

Current Project

UuT Defined

[✓] C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Project

Execute before Test

Execute after Test

C:\CopyReports.exe

Close Test Bench Connection

[✓] when aborting execution on testcase error (abort on error flag must be set)

[✓] on stopping by user

- Click **OK**.

  The application selected is launched once the test has been run.

Selecting a Report Generator

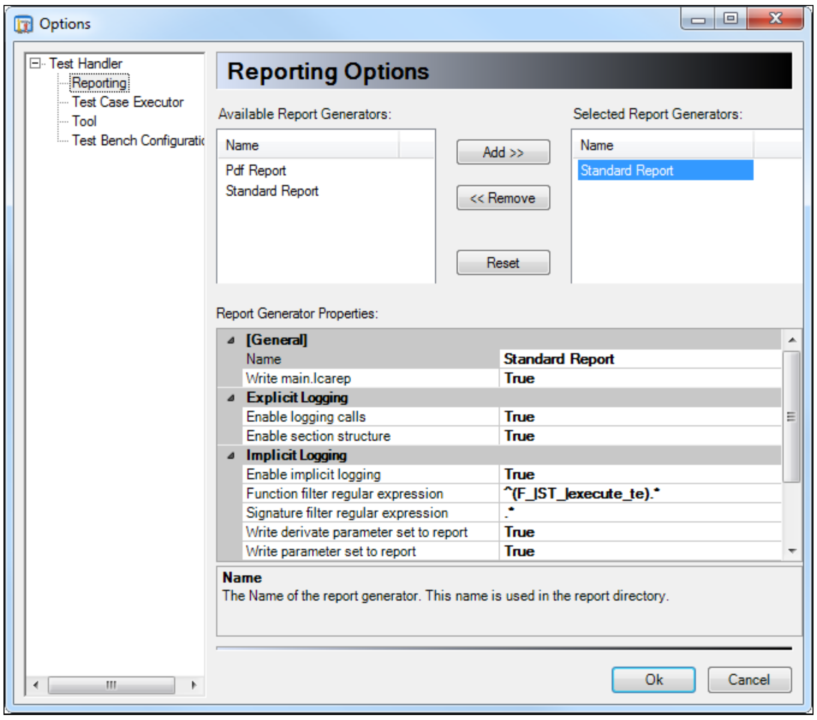
To document the test in a test report, a report generator has to be specified. The "Standard Report" is currently available for this purpose: it stores the required information in XML files.

To select settings for the report generator

- Select **Tools** → **Options** from the main menu of the Test Handler.

  The "Options" window opens.

- Select "Test Handler\Reporting" from the left-hand side of the window.



Reporting Options

Available Report Generators:

Name
Pdf Report
Standard Report

Selected Report Generators:

Name
Standard Report

Report Generator Properties:

[General]	
Name	**Standard Report**
Write main.lcarep	**True**
Explicit Logging	
Enable logging calls	**True**
Enable section structure	**True**
Implicit Logging	
Enable implicit logging	**True**
Function filter regular expression	**^(F_IST_|execute_te).***
Signature filter regular expression	**.***
Write derivate parameter set to report	**True**
Write parameter set to report	**True**

Name
The Name of the report generator. This name is used in the report directory.

- If there is no report generator selected in the "Selected Report Generators" field, select it in the "Available Report Generators" field.
- Add this by pressing the **Add>>** button.

  Depending on the report generator selected, you can now set different options for it in the "Report Generator Properties" field. These are described below.
- Once you have defined all settings, quit this window by pressing **OK**.

Section/Option	**Meaning**
[General]	
Name	The name of the selected report generator
Write main.lcarep	Specifies whether an overview file (`main.lcarep`) is to be created ([True|False])
Explicit Logging	
Enable logging calls	Specifies whether function calls are logged ([True|False])
Enable section structure	Specifies whether section calls are logged ([True|False])
Implicit Logging	
Enable implicit logging	Enable/disable implicit logging
Function filter regular expression	Uses a regular expression to specify which function calls are logged. The default setting is `^(F_|ST_|execute_te).*`, i.e. all function calls starting with F_ or ST_ or execute_te are logged.
Signature filter regular expression	Uses a regular expression to specify which signature calls are logged. The default setting is .*, i.e. all signature calls are logged.
Write derivate parameter set to record	Specifies whether derivates of parameter sets are logged ([True|False])
Write parameter set to record	Specifies whether parameter sets are logged ([True|False])
Write TTCN structure to report	Specifies whether the test run is written to the report.
Plot Display Settings	
PlotHeight	Plot height (in pixels)
PlotWidth	Plot width (in percent)

Tab. 4-2 Options for the Report Generator

To run, pause and stop a test run

- To start the test run, select **Execution** → **Start Test Run**.

or

- Click the **Run** icon in the toolbar.

  The test run is launched.

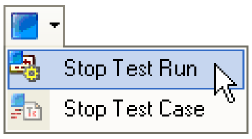
- To pause the test run, select **Execution** → **Pause Test Run**.

or

- Click the **Pause** icon in the toolbar.

  The test run is paused, but not stopped, and can be continued using **Execution** → **Run**.
- To stop the test run, select **Execution** → **Stop Test Run**.

or



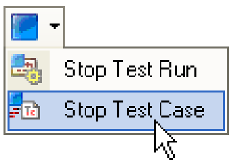
- Click the **Stop** → **Stop Test Run** icon in the toolbar.

  The test run is stopped once the current test case has been executed.

To stop the execution of individual test cases

- If you want to stop (= cancel) an individual test case and not the entire test run, select **Execution** → **Stop Test Case**.

or



- Click the **Stop** → **Stop Test Case** icon in the toolbar.

  The current test case is stopped and the subsequent test cases are run.

Clicking **Stop Test Case** triggers a “StopTestCase” event handled by the Event Handler. This means the Event Handler carries out the following actions in the order listed:

- Calls the “StopTestCase” method provided by the ATCL,
- Issues a short note that the test case has been ended,
- Sets the verdict “inconc”

and

- Stops running the test case.

Initially, the “StopTestCase” method provided by the ATCL does nothing. However, it is possible to run specific actions before the relevant test case is stopped by overriding the method.

In the following C# example, the string “Test case stopped” is issued ten times (with a certain gap between each case):

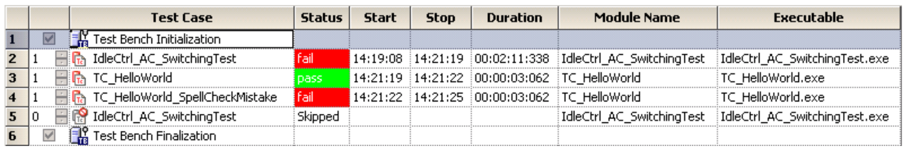
```
public override void StopTestCase()
{
   int count = 0;
   while (count != 10)
   {
   Reporting.SetText(0,0,"Test case stopped",0);
   Thread.Sleep(250);
   count++;
   }
}
```

Note

If the "StopTestCase" method provided by the ATCL is not overriden by the test case and the test case is cancelled with Stop Test Case, the test bench remains in the state the test case has established/changed up to this point. Open tools are closed by the application. Subsequent test cases are run on the modified test bench configuration! This may lead to modified test results.

Tracing Test Execution

The progress of the test can be traced in the tabular view of Test Handler (Fig. 4-17).



		Test Case	Status	Start	Stop	Duration	Module Name	Executable
1	✓	Test Bench Initialization						
2	1	IdleCtrl_AC_SwitchingTest	fail	14:19:08	14:21:19	00:02:11:338	IdleCtrl_AC_SwitchingTest	IdleCtrl_AC_SwitchingTest.exe
3	1	TC_HelloWorld	pass	14:21:19	14:21:22	00:00:03:062	TC_HelloWorld	TC_HelloWorld.exe
4	1	TC_HelloWorld_SpellCheckMistake	fail	14:21:22	14:21:25	00:00:03:062	TC_HelloWorld	TC_HelloWorld.exe
5	0	IdleCtrl_AC_SwitchingTest	Skipped				IdleCtrl_AC_SwitchingTest	IdleCtrl_AC_SwitchingTest.exe
6	✓	Test Bench Finalization						

Fig. 4-17 The Tabular View during Test Execution

The meaning of the individual columns has already been described in the section "The Tabular View of Test Handler" on page 111.

4.5.8 Running the Test Handler from the Command Line

The Test Handler can also be run from the Windows command line. The syntax is as follows:

```
THCMD.exe <projectfile> -u [<UuT name>]
[-c <Sequence name>] [-t <TestRunConfiguration name>]
```

The arguments are:

- `<projectfile>`

  Specifies the project or batch file.
- `-u <UuT name>`

  The project's UuT to be run (not valid with batch projects).
- `-c <Sequence name>`

  Specifies the sequence to be run (optional). If a sequence name was specified, the default Test Run Configuration of this sequence is run.
- `-t <TestRunConfiguration name>`

  Specifies the Test Run Configuration to be run (optional).

No other arguments are allowed.

The exit codes have the following meanings:

Exit Code	Meaning
0	Success
1	Error
2	Application already running
3	Error loading project

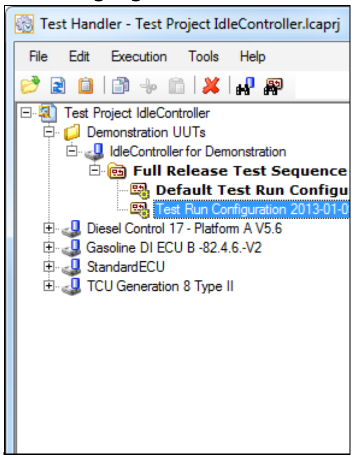
4.5.9 Test Reports

After every test execution, reports in the form of XML files are created and stored in the selected directory (see "Specifying the Report Directory" on page 117) in the project folder.

The following files and directories are created:

- One report directory each (`Report YYYY-MM-DD HHMMSS`) per Test Run Configuration executed.
- A further directory is created in this - `\Standard Report`
  It contains the main report file `main.lcarep` and other subdirectories with the report files of the individual test cases.

Every test execution generates a new report folder in the project and a new entry under the current Test Run Configuration in the Project Explorer as shown in the following figure:



Test Project IdleController
Demonstration UUTs
IdleController for Demonstration
Full Release Test Sequence
Default Test Run Configu
Test Run Configuration 2013-01-0
Diesel Control 17 - Platform A V5.6
Gasoline DI ECU B -82.4.6.-V2
StandardECU
TCU Generation 8 Type II

Fig. 4-18 Report Entries in the Project Explorer

To view test reports

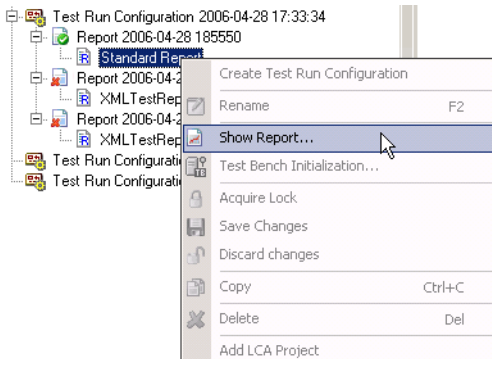
There is an entry for each test report under the relevant Test Run Configuration.

- To open the test report, double-click "Standard Report" below "Report YYYY-MM-DD HHMMSS".

or

- Select "Standard Report".

- Select **Show Report** from the shortcut menu.



The Report Viewer is launched and the test report displayed in accordance with the settings specified by the Report Manager.

For more details on the role of the Report Manager, refer to the section "The Tasks of the Report Manager" on page 134.

4.5.10 Working with Batch Projects

If the Test Handler wants to run several Test Run Configurations of a LABCAR-AUTOMATION project automatically, it is possible to create a batch project.

Batch Projects

A batch project consists of a group of Test Run Configurations of a LABCAR-AUTOMATION project.

This kind of batch project can be created, saved, loaded and modified by the Test Handler in the application of the same name.

Several batch projects can refer to one and the same LABCAR-AUTOMATION project which can be modified while the batch projects are open. If the changes are saved, the Test Handler is informed of this and the batch project is reloaded.

User Interface

In addition to the "Test Run" tab, the user interface of the Test Handler has an additional tab, called "Batch Run", when dealing with a batch project. It shows the selected Test Run Configurations. Test Run Configurations can be added and their sequence modified using drag-and-drop or via the shortcut menu.

Consistency Check

If a batch project is loaded and run, a check takes place automatically to see whether all Test Run Configurations of the batch project actually exist in the referenced LABCAR-AUTOMATION project.

If the batch project contains Test Run Configurations which have been deleted from the test project, these are marked in red in the "Batch Run" tab and have to be deleted from the batch project.

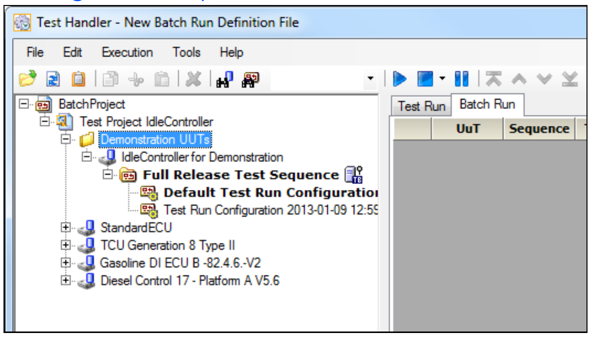
To create a batch project

- From the main menu of the Test Handler, select **File** → **New Batch Project**.

  A file selector window opens from which you can select the LABCAR-AUTOMATION project (*.lcaprj) which contains the corresponding sequences.

- Select the project and click **OK**.

  The project with all its UuTs, sequences and Test Run Configurations opens.


Test Handler - New Batch Run Definition File
File
Edit
Execution
Tools
Help
BatchProject
Test Project IdleController
Demonstration UUTs
IdleController for Demonstration
Full Release Test Sequence
Default Test Run Configuratio
Test Run Configuration 2013-01-09 12:5
StandardECU
TCU Generation 8 Type II
Gasoline DI ECU B -82.4.6.-V2
Diesel Control 17 - Platform A V5.6
Test Run
Batch Run
UuT
Sequence

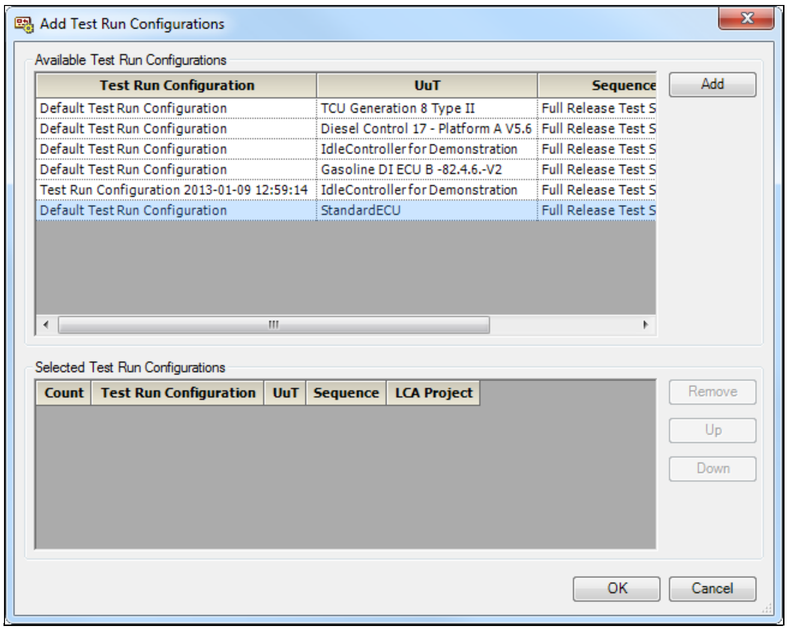
  In addition, in the right-hand side of the window, a new "Batch Run" tab is added in which the Test Run Configurations for the batch run are listed.

- If necessary, create additional Test Run Configurations.

To add Test Run Configurations to a batch run

- Right-click the "Batch Run" tab and select **Append Test Run Configurations**.

  The "Add Test Run Configurations" window opens.
- Select the required Test Run Configurations (multiple selection with <Shift> or <Ctrl> is possible) and click **Add**.


Add Test Run Configurations
Available Test Run Configurations
Add
Selected Test Run Configurations
Remove
Up
Down
OK
Cancel

Test Run Configuration	UuT	Sequence
Default Test Run Configuration	TCU Generation 8 Type II	Full Release Test S
Default Test Run Configuration	Diesel Control 17 - Platform A V5.6	Full Release Test S
Default Test Run Configuration	IdleController for Demonstration	Full Release Test S
Default Test Run Configuration	Gasoline DI ECU B -82.4.6.-V2	Full Release Test S
Test Run Configuration 2013-01-09 12:59:14	IdleController for Demonstration	Full Release Test S
Default Test Run Configuration	StandardECU	Full Release Test S

Count	Test Run Configuration	UuT	Sequence	LCA Project

Note

Every column of the window can be sorted alphabetically in standard or reverse order by clicking on the column name.

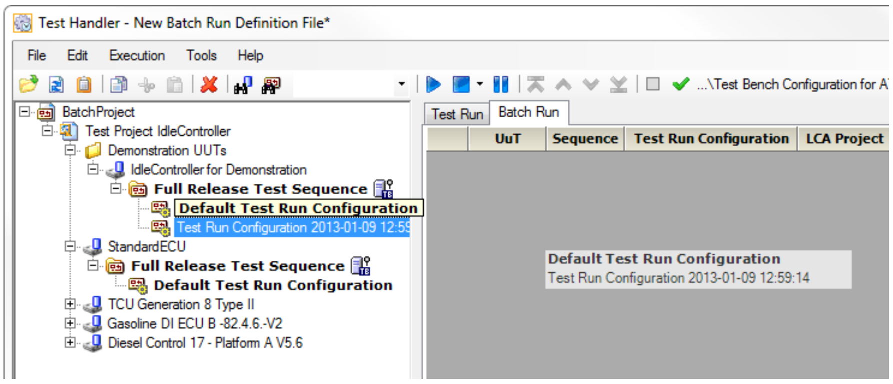
The selected Test Run Configurations are added to the "Selected Test Run Configurations" list.

- To remove an item from the list of available test cases, select it and click **Remove**.
- Click **OK**.

  The selected Test Run Configurations are added to the "Batch Run" tab.

or

- Move the corresponding Test Run Configurations by drag-and-drop to the "Batch Run" tab. You can select multiple Test Run Configurations in the project window.


Test Handler - New Batch Run Definition File*
File Edit Execution Tools Help
...\Test Bench Configuration for A
BatchProject
Test Project IdleController
Demonstration UUTs
IdleController for Demonstration
Full Release Test Sequence
Default Test Run Configuration
Test Run Configuration 2013-01-09 12:59
StandardECU
Full Release Test Sequence
Default Test Run Configuration
TCU Generation 8 Type II
Gasoline DI ECU B -82.4.6.-V2
Diesel Control 17 - Platform A V5.6
Test Run
Batch Run
UuT
Sequence
Test Run Configuration
LCA Project
Default Test Run Configuration
Test Run Configuration 2013-01-09 12:59:14

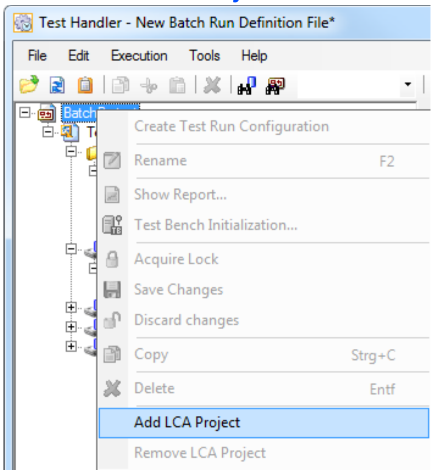
The selected Test Run Configurations are inserted in the list in the "Batch Run" tab.

- Save the batch file with **File** → **Save As**.

To remove and add projects

You can also add other LABCAR-AUTOMATION projects to your batch project.

- Select the batch project from the left-hand side of the Test Handler.
- Select **Add LCA Project** from the shortcut menu.


Test Handler - New Batch Run Definition File*
File Edit Execution Tools Help
Batch
Create Test Run Configuration
Rename F2
Show Report...
Test Bench Initialization...
Acquire Lock
Save Changes
Discard changes
Copy Strg+C
Delete Entf
Add LCA Project
Remove LCA Project

A file selector window opens.

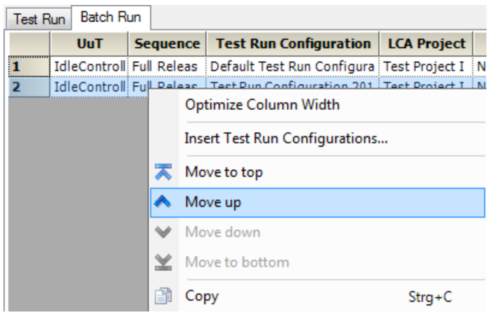
- Select the project file and click **OK**.

The project is added to your batch project. You can now add Test Run Configurations from the new project to your batch run definition.

- To remove a LABCAR-AUTOMATION project again, select **Remove LCA Project** from the shortcut menu above.

To change the order of the Test Run Configurations in the batch project

- To move a Test Run Configuration in the list, select one of the options in the shortcut menu.

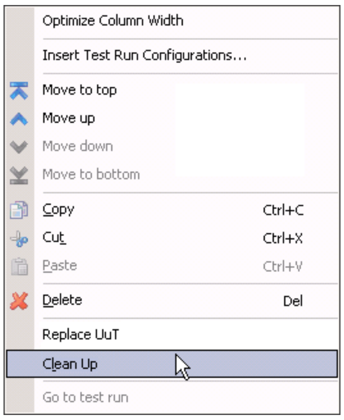


To delete Test Run Configurations from the list

If your batch run definition contains Test Run Configurations of a removed project, these are indicated in red in the "Batch Run" tab. You can remove these using the shortcut menu.

- To delete a single Test Run Configuration from the batch project, select **Delete** from the shortcut menu.

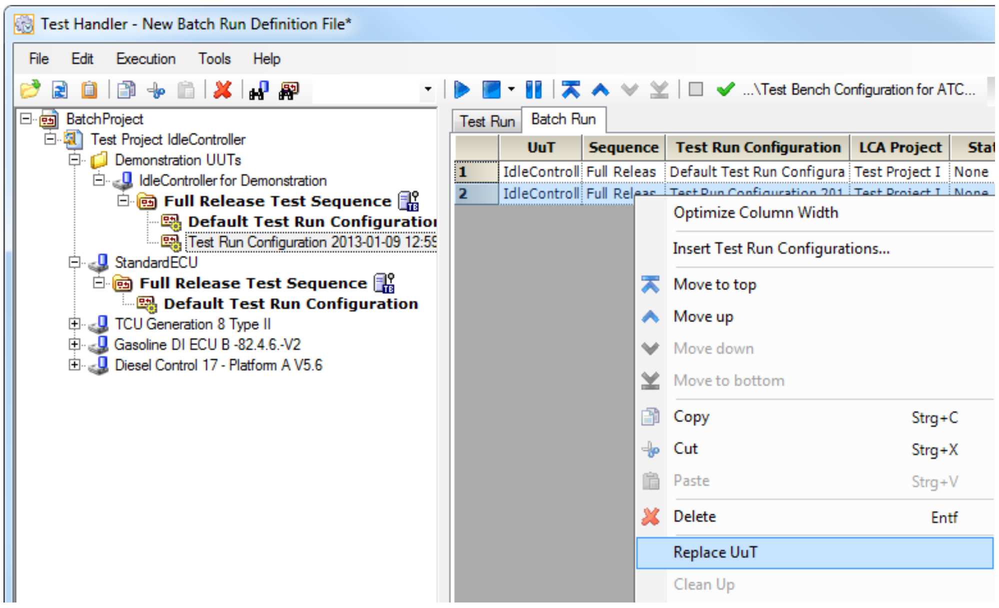
- To delete all invalid (red) Test Run Configurations from the Batch project, select **Clean Up** from the shortcut menu.



To exchange UuTs in the batch run definition

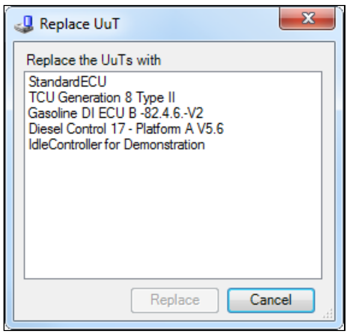
You can exchange UuTs of the batch run definitions for others of the same UuT Group. Proceed as follows:

- Select the UuT to be exchanged in the "Batch Run" tab.
- Select **Replace UuT** from the shortcut menu.


Test Handler - New Batch Run Definition File*
File Edit Execution Tools Help
...\Test Bench Configuration for ATC...
BatchProject
Test Project IdleController
Demonstration UUTs
IdleController for Demonstration
Full Release Test Sequence
Default Test Run Configuration
Test Run Configuration 2013-01-09 12:59
StandardECU
Full Release Test Sequence
Default Test Run Configuration
TCU Generation 8 Type II
Gasoline DI ECU B -82.4.6.-V2
Diesel Control 17 - Platform A V5.6
Test Run
Batch Run
Optimize Column Width
Insert Test Run Configurations...
Move to top
Move up
Move down
Move to bottom
Copy Strg+C
Cut Strg+X
Paste Strg+V
Delete Entf
Replace UuT
Clean Up

	UuT	Sequence	Test Run Configuration	LCA Project	Stat
1	IdleControll	Full Releas	Default Test Run Configura	Test Project I	None
2	IdleControll	Full Releas	Test Run Configuration 201	Test Project I	None

A window opens containing all UuTs of the UuT Group of this project to which the UuT to be replaced also belongs.



StandardECU
TCU Generation 8 Type II
Gasoline DI ECU B -82.4.6.-V2
Diesel Control 17 - Platform A V5.6
IdleController for Demonstration

- Select the new UuT and click **Replace**.

All Test Run Configurations in the "Batch Run" tab are removed and replaced by Test Run Configurations of the same test sequence, but of the new UuT. The parameters of the old and of the new Test Run Configuration (TB Init, TB Finalize, Iteration Number) must correspond to each other. If this is not the case for the test sequence of the new UuT, a corresponding Test Run Configuration is created automatically and added to the project.

To run a batch project

- To run the batch project shown in the "Batch Run" tab, select **Execution** → **Start Batch** from the main menu.

or

- Press <F5>.

or

- Click the **Start** icon.

  The batch project starts.

- To pause the batch project, select **Execution** → **Pause Batch** from the main menu.

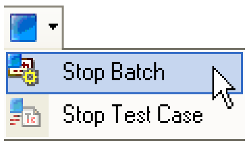
or

- Click the **Pause** icon.

  The batch project is paused and can be continued with **Execution** → **Start Batch**.

- To cancel the batch project, select **Execution** → **Stop Batch** from the main menu.

or



- Click the **Stop** → **Stop Batch** icon in the toolbar.

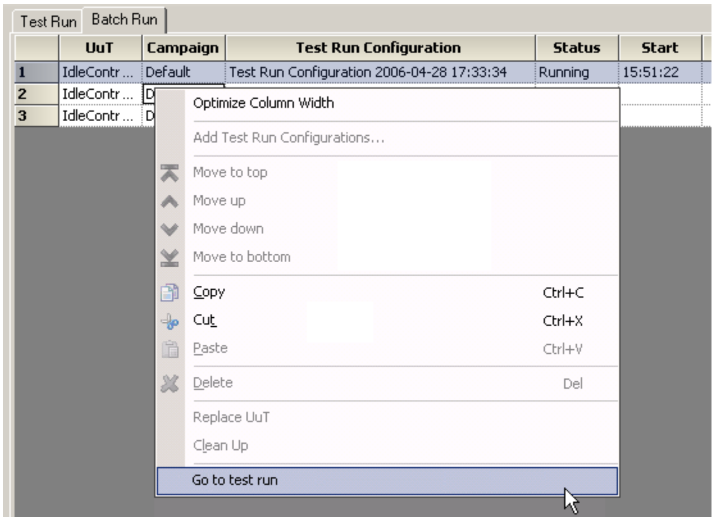
  The batch run is stopped once the current test case has been executed.

Use **Stop Test Case** to stop the execution of individual test cases (see "To stop the execution of individual test cases" on page 123).

To skip test runs

You can skip test runs which have not been called during a running batch project and then continue with a selected test run.

- Select the Test Run Configuration you want to continue with from the "Batch Run" tab.
- Select **Interrupt current test run and go to selected test run** from the shortcut menu.

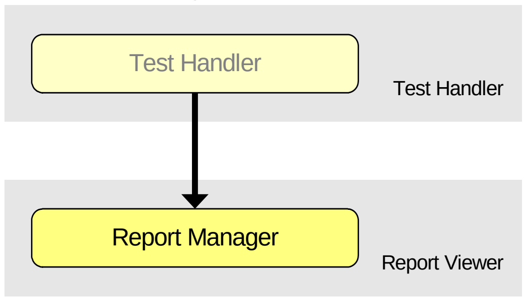


	UuT	Campaign	Test Run Configuration	Status	Start
1	IdleContr ...	Default	Test Run Configuration 2006-04-28 17:33:34	Running	15:51:22
2	IdleContr ...	D			
3	IdleContr ...	D			

The current test run is interrupted and the batch project is continued with the selected test run.

4.6 The Tasks of the Report Manager

The Report Manager is responsible for displaying test reports saved in XML format which were generated when a test sequence was executed.



The views of the test reports can be modified to perfectly suit your particular requirements by using different templates and stylesheets and defining filter criteria.

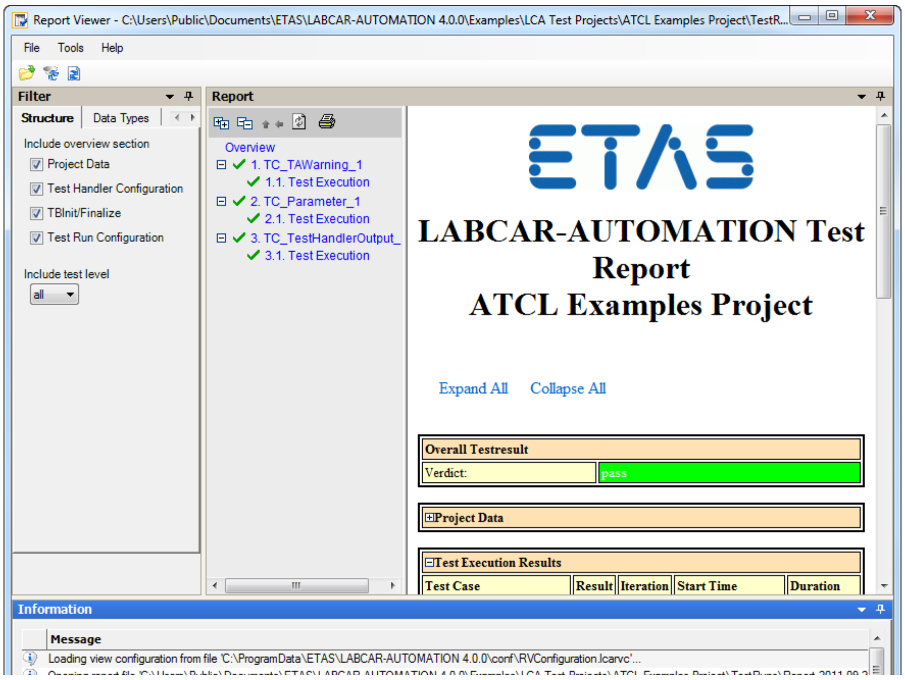
The Report Manager has the following tasks:

- Selection of the `main.lcarep` file which contains all test reports of the executed test sequence, including all the information on the project and the test bench used.
- Selection of the XLST stylesheets for displaying the test report in the HTML Browser
- Definition of the filter criteria
- Display of the test results
- Saving the generated view

Below, you will find a description of the user interface in which the Report Manager executes its tasks.

4.6.1 The User Interface of the Report Manager

Fig. 4-19 shows the user interface of the Report Manager with an open Test Report.



Report Viewer - C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\LCA Test Projects\ATCL Examples Project\TestR...

Include overview section
☑ Project Data
☑ Test Handler Configuration
☑ TBInit/Finalize
☑ Test Run Configuration

Include test level: all

Overview
- ✓ 1. TC_TAWarning_1
  - ✓ 1.1. Test Execution
- ✓ 2. TC_Parameter_1
  - ✓ 2.1. Test Execution
- ✓ 3. TC_TestHandlerOutput_
  - ✓ 3.1. Test Execution

LABCAR-AUTOMATION Test Report
ATCL Examples Project

Overall Testresult	
Verdict:	pass

⊞Project Data

⊟Test Execution Results

Test Case	Result	Iteration	Start Time	Duration

Loading view configuration from file 'C:\ProgramData\ETAS\LABCAR-AUTOMATION 4.0.0\conf\RVConfiguration.lcarvc'...

Fig. 4-19 The User Interface of the Report Manager

The user interface consists of the following elements:

The "Report" Window

The generated report is displayed in this window.

The "Filter" Window

This window contains the possible settings of the filter (see "Filtering Test Report Data" on page 138) and the button for renewing a view generated "on-the-fly" (see "Displaying Reports during Test Execution" on page 146).

The "Information" Window

This window contains information on processing the report files and messages on any errors which may have occurred.

The "File" Menu

The "File" menu is used to save and open test reports, views and view configurations.

This menu contains the following items:

- **Open Report**

  Opens a test report (see "To open a test report" on page 136).
- **Open Report View**

  Opens a view (see "To load a view" on page 143).

- **Save Report View**

  Saves a view (see "To save a view" on page 143).
- **Save Report View as**

  Saves a view under a specific name (see "To save a view" on page 143).
- **Open View Configuration**

  Opens a view configuration (see "To open a view configuration" on page 144).
- **Save View Configuration As**

  Saves a view configuration under a specific name (see "To save a view configuration" on page 144).
- **Exit**

  Ends the Report Manager.

The "Tools" Menu

Adaptations such as selecting the stylesheet, customer logo used etc., are made in the "Tools" menu.

This menu contains the following item:

- **Options**

  This is where settings for report display are made (see "Report Manager Settings" on page 144).

The "Help" Menu

- **Help → User's Guide**

  Opens the English User's Guide (PDF).
- **Help → System Info**

  Opens an information window containing details of the current versions of the components of LABCAR-AUTOMATION 4.2.3.
- **Help → About**

  Opens a window containing general information.

4.6.2 Displaying Test Reports (View)

A test report is generated during test sequence in the form of XML. The Report Manager generates a specific view of the test report data (report view) when it "opens" this test report.

The content and appearance of such a report view is determined by the data available in the XML files, the selected filter options and the XSLT stylesheet used.

To open a test report

To generate a report view from the available XML files, proceed as follows:

- Select **File → Open Report**.

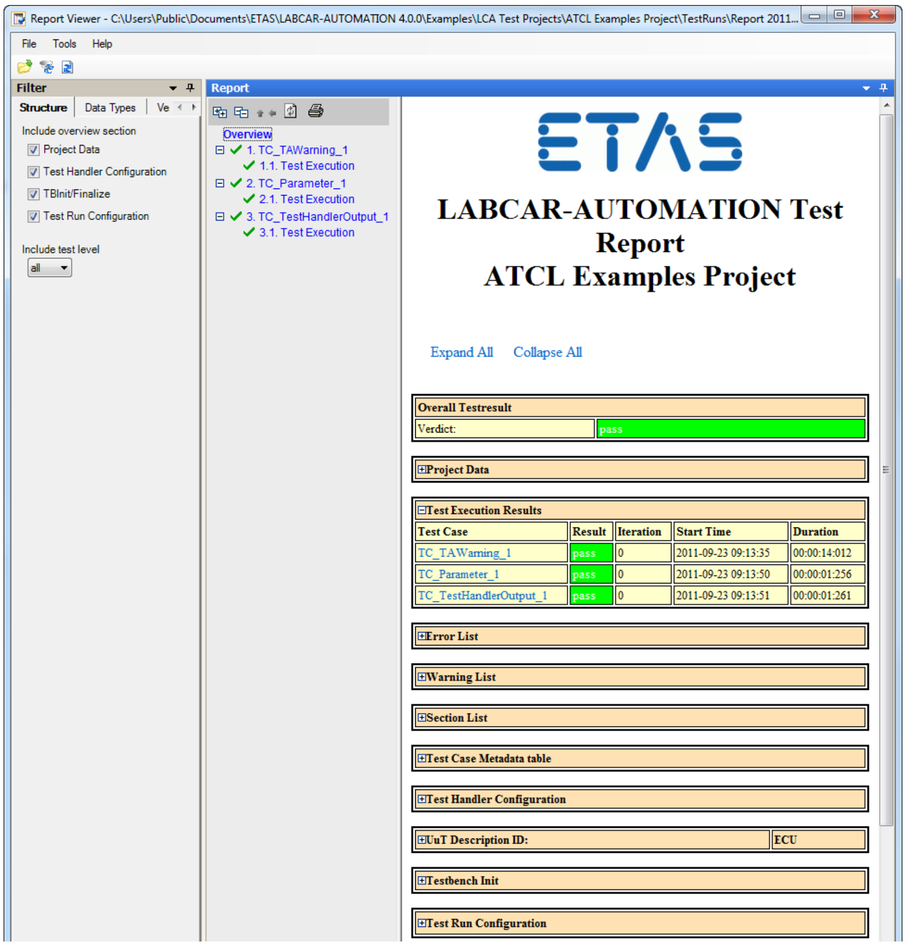
  A file selector window opens.
- Select the `main.lcarep` file in the directory which contains the data for the test report to be displayed.

- Click **OK**.

  The report view with the default stylesheet and default filter settings is created in the "Report" window (see Fig. 4-20 on page 137).

The "Overview" Section

A test report starts with an overview of the different aspects of the test project run.


Report Viewer - C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\LCA Test Projects\ATCL Examples Project\TestRuns\Report 2011...

Filter — Structure | Data Types | Ve

Include overview section
- [✓] Project Data
- [✓] Test Handler Configuration
- [✓] TBInit/Finalize
- [✓] Test Run Configuration

Include test level: all

Report

Overview
- ✓ 1. TC_TAWarning_1
  - ✓ 1.1. Test Execution
- ✓ 2. TC_Parameter_1
  - ✓ 2.1. Test Execution
- ✓ 3. TC_TestHandlerOutput_1
  - ✓ 3.1. Test Execution

LABCAR-AUTOMATION Test Report
ATCL Examples Project

Expand All   Collapse All

Overall Testresult	
Verdict:	pass

⊞Project Data

⊟Test Execution Results

Test Case	Result	Iteration	Start Time	Duration
TC_TAWarning_1	pass	0	2011-09-23 09:13:35	00:00:14:012
TC_Parameter_1	pass	0	2011-09-23 09:13:50	00:00:01:256
TC_TestHandlerOutput_1	pass	0	2011-09-23 09:13:51	00:00:01:261

⊞Error List

⊞Warning List

⊞Section List

⊞Test Case Metadata table

⊞Test Handler Configuration

⊞UuT Description ID: ECU

⊞Testbench Init

⊞Test Run Configuration

Fig. 4-20 The Structure of a Test Report

The "Overview" section of a test report consists of the following sections:

1. Overall results ("Overall Testresult")
2. Project data ("Project Data") *
3. Results of the test cases run ("Test Execution Results")
4. List of sections with warnings or errors ("Errorlist")
5. The metadata of the test cases ("Test Case Metadata table")
6. Configuration data ("Test Handler Configuration") *
7. UuT data ("UuT Description ID")
8. Test bench configuration ("Testbench Init") *
9. Test run configurations ("Test Run Configuration") *

10. Test bench configuration ("Test Bench Configuration")

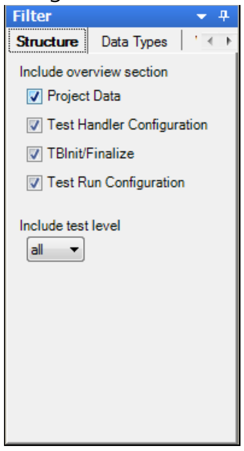
The display of the sections shown with an * can be suppressed using a filter (see "Filter for the Entire Report Structure" on page 139).

The Sections for the Test Cases

After the "Overview" section, the individual test cases are evaluated. A sample evaluation can be found in the tutorial under "To evaluate a test report" on page 206.

4.6.3 Filtering Test Report Data

To adapt the view of the data of a report, there are several filter options which can be selected in the "Filter" window. The different filter types can be selected using the tab of this window.



Include overview section

- [✓] Project Data
- [✓] Test Handler Configuration
- [✓] TBInit/Finalize
- [✓] Test Run Configuration

Include test level

all

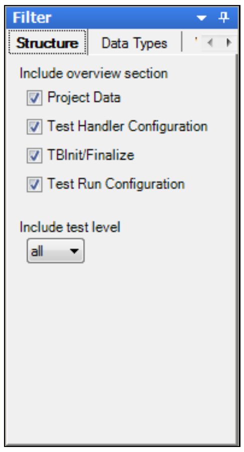
Fig. 4-21 The "Filter" Window

After changing filter settings, the display in the report view is updated clicking the **Reload Document (with current filter settings)** icon.

The following groups of filter options can be selected:

- "Filter for the Entire Report Structure" on page 139
- "Filter over "Data Types"" on page 140
- "Filter over the Test Result ("Verdicts")" on page 140
- "Filter over "Text Classes"" on page 141
- "Filter over "Report Level"" on page 142

Filter for the Entire Report Structure



Include overview section

- [✓] Project Data
- [✓] Test Handler Configuration
- [✓] TBInit/Finalize
- [✓] Test Run Configuration

Include test level: all

Fig. 4-22 The ”Structure” Tab of the ”Filter” Window

In the ”Structure” tab, two ”global” settings on the level of detail of the report can be made:

- Include overview section

  The scope of the meta data displayed (see "The Structure of a Test Report" on page 137):

  – Project Data

  – Test Handler Configuration

  – TBInit/Finalize

  – Test Run Configuration Table

- Include test levels

  The hierarchical level of detail, i.e. which hierarchy levels are displayed.

  – all levels

    All hierarchy levels are displayed.

  – 1 .. 9

    Number of all hierarchy levels displayed (”4” means that levels 1 to 4 are displayed).

Filter over "Data Types"

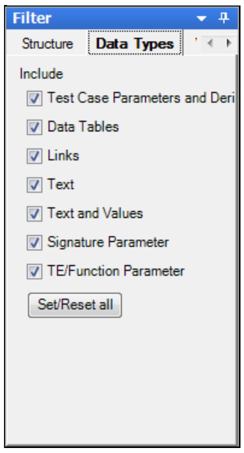


Fig. 4-23 The "Data Types" tab of the "Filter" Window

In this tab, the display of a report can be filtered with reference to certain data types.

Available data types are:

- Test Case Parameters and Derivatives
- Data Tables
- Links
- Text
- Text and Values
- Signature Parameters
- TE/Function Parameter

You can activate or deactivate all options at the same time with **Set/Reset all**.

Filter over the Test Result ("Verdicts")

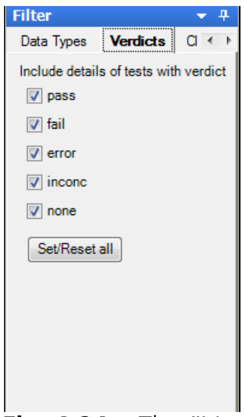


Fig. 4-24 The "Verdicts" Tab of the "Filter" Window

This filter refers to the section of a report in which the individual Test Run Configurations are evaluated. The display of a Test Run Configuration in the view can be made dependent on the result of the execution.

Available options are:

- pass
- fail
- error
- inconc
- none

You can activate or deactivate all options at the same time with **Set/Reset all**.

Filter over "Text Classes"

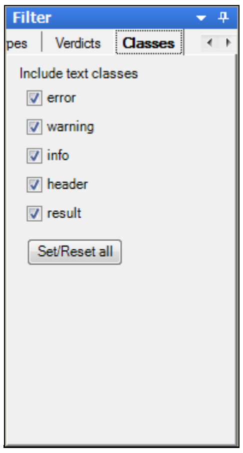


Fig. 4-25 The "Classes" Tab of the "Filter" Window

This filter refers to the parameter "TextClass" of TTCN protocol functions such as "LogTextAndValue", "LogText", "LogLinkToFile", etc.

The display of certain text classes can be selected or excluded here.

Available options are:

- error
- warning
- info
- header
- result

You can activate or deactivate all options at the same time with **Set/Reset all**.

Filter over "Report Level"

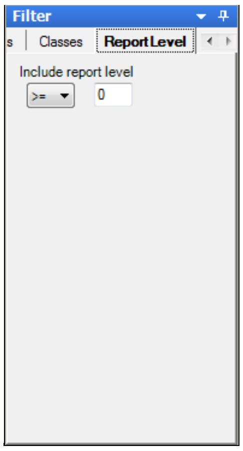


Fig. 4-26 The "ReportLevel" Tab of the "Filter" Window

This filter refers to the attribute "ReportLevel" of TTCN protocol functions such as "LogTextAndValue", "LogText", "LogLinkToFile", etc.

The following comparing operations are possible:

- <=

  Smaller than or equal to the specified level.
- =

  Equal to the specified level.
- >=

  Greater than or equal to the specified level.
- <>

  Not equal (smaller or greater than) the specified level.

A certain level ("equal") or an upper ("less or equal") or lower ("greater or equal") limit for the levels to be displayed can be selected.

4.6.4 Reports, Report Views and View Configurations

Test reports are stored as XML files – when a test report is opened in Report Viewer, it is displayed as an HTML file. A range of files such as stylesheets, logos etc. are required for this purpose. These are defined in Report Viewer using **Tools → Options** (see "Report Manager Settings" on page 144).

Such HTML representations of a test report can be saved in Report Viewer as report views (see "Saving and Loading Views ("Report Views")" on page 143). The test report is saved in the form of an HTML file and can be opened with every HTML browser.

As described in "Filtering Test Report Data" on page 138, the information displayed in a report can be filtered to a certain extent. In turn, these filter settings can be saved in Report Viewer as view configurations (with the file extension "lcarvc") and reloaded (see "Saving and Loading View Configurations" on page 143).

4.6.5 Saving and Loading Views (”Report Views”)

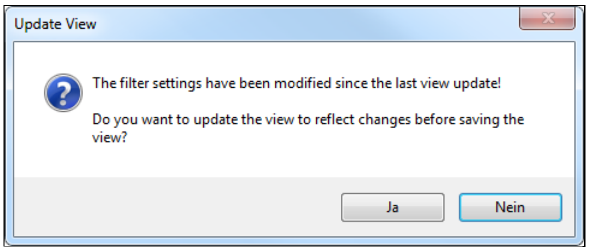
Saving a report view means saving all components of this view such as pictures, CSS files, data etc. so that the view can be repeated at any time. In addition, general configuration data is saved (what is referred to as the view configuration), to which e.g. filter options belong (see "Saving and Loading View Configurations" on page 143).

The XSLT stylesheet provided generates HTML files which can be displayed with the Internet Explorer 6.0[1] - Report Viewer is not necessary for this.

To save a view

- If you are saving the view for the first time, select **File** → **Save Report View As**.

  If you have changed filter settings, but have not executed an **Reload Document (with current filter settings)** since then, you are asked whether you want to execute an update of the view before saving (**Yes**) or not (**No**).


Update View

The filter settings have been modified since the last view update!

Do you want to update the view to reflect changes before saving the view?

- Select the relevant button.

  A directory selector window opens.
- Click **OK**.

  The view is saved as an HTML file.

  If you want to save a view (e.g. after modifying it) under its old name, select **File** → **Save Report View**.

To load a view

- To load a saved view, select **File** → **Open Report View**.
- Select the required file in the file selector window.

  The relevant HTML file opens.

4.6.6 Saving and Loading View Configurations

A view configuration includes the following settings:

- The filter options selected in the ”Filter” window (see "Filtering Test Report Data" on page 138)

[1] The stylesheet has been developed for this browser – it may well work with other browsers.

- The customer logo selected (see "Report Manager Settings" on page 144)

View configurations have the file extension "lcarvc".

To save a view configuration

- If you are saving the view configuration for the first time, select **File** → **Save View Configuration As**.
- Enter a file name.
- Click **OK**.
- If you want to save a view configuration (e.g. after modifying it) under its old name, select **File** → **Save View Configuration**.

To open a view configuration

- To load a saved view configuration, select **File** → **Open View Configuration**.

4.6.7 Report Manager Settings

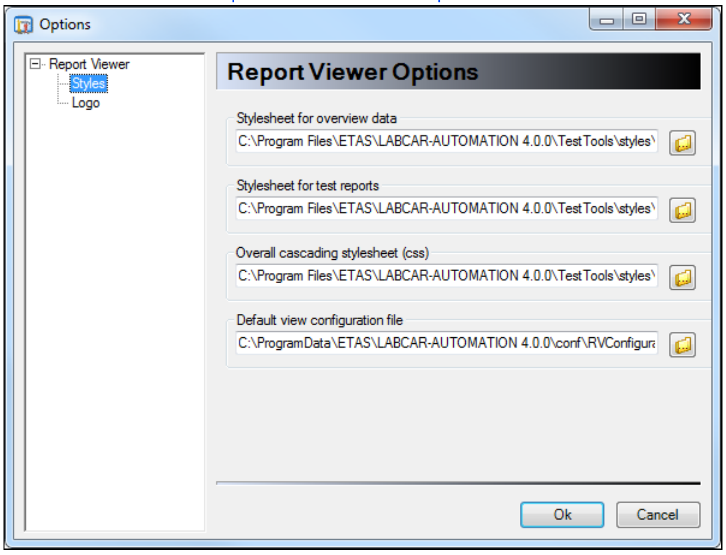
The following settings of the Report Manager can be influenced:

- The XSLT stylesheets for converting XML files into HTML files.
- The file with the customer logo and the size and alignment for the display of the logo at the start of the test report.

To define settings

- Select **Tools** → **Options**.

  The "Options" window opens.



Stylesheet for overview data: C:\Program Files\ETAS\LABCAR-AUTOMATION 4.0.0\TestTools\styles\

Stylesheet for test reports: C:\Program Files\ETAS\LABCAR-AUTOMATION 4.0.0\TestTools\styles\

Overall cascading stylesheet (css): C:\Program Files\ETAS\LABCAR-AUTOMATION 4.0.0\TestTools\styles\

Default view configuration file: C:\ProgramData\ETAS\LABCAR-AUTOMATION 4.0.0\conf\RVConfigura

- Select "Styles" on the left-hand side of the window.

- Select the required stylesheets using the relevant folder icons.

  The following styles can be specified:

  – Stylesheet for overview data

    The stylesheet for the "Overview" section (see Fig. 4-20 on page 137).

  – Stylesheet for test reports

    The stylesheet for the reports of the individual Test Run Configurations.

  – Stylesheet for link list frame

    The stylesheet for the list in the left-hand part of Fig. 4-20 on page 137.

  – Overall cascading stylesheet

    The Cascading Style Sheet for the overall display.

  – Default view configuration file

    A default view configuration file (see section 4.6.6 on page 143).

Note

All paths must be specified relative to the installation directory, e.g. to <drive>:\Program Files\ETAS\LABCAR-AUTOMATION 4.2\.

- Select the entry "Logo" from the tree view.

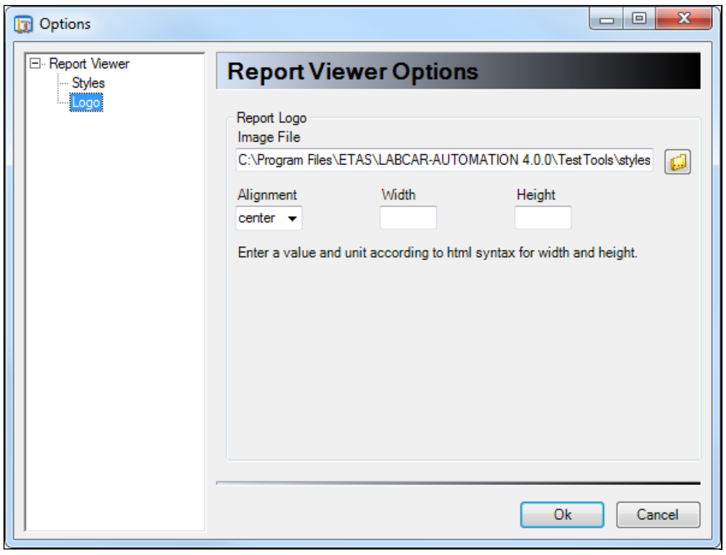


Image File
C:\Program Files\ETAS\LABCAR-AUTOMATION 4.0.0\TestTools\styles

Alignment: center
Width:
Height:

Enter a value and unit according to html syntax for width and height.

- Select the file with the logo desired.
- Specify "Alignment" and dimensions ("Width", "Height").

The selected settings can be saved in the form of a view configuration and if necessary reloaded (see "Saving and Loading View Configurations" on page 143).

4.6.8 Displaying Reports during Test Execution

As test reports are already written while the test is being executed, data already available can be displayed during test execution.

Whenever the user selects a folder with XML test report data (see "To open a test report" on page 136), all data available up to this point is displayed.

This display is not, however, automatically kept up-to-date – to display the most recent data set, click the **Refresh Document** icon in the toolbar.

5 Configuration Files and their Editors

This chapter describes the configuration files which are managed by the Test Bench Configuration Manager (TBCM).

The TBCM has editors at its disposal, the use of which is also described in this chapter.

This chapter includes information on the following topics:

- "The Test Bench Configuration File Editor" on page 148

  The Test Bench Configuration File mainly contains the mapping of port instances used in the test cases to specific tool instances.
- "The Tool Configuration File Editor" on page 163

  A Tool Configuration File (TCF) contains configurations for a test bench tool required to run the test project.
- "The Switch Definition File Editor" on page 174

  The Switch Definition File contains the definition of all switches to be used.
- "The UuT List Editor" on page 186

  The UuT List Editor is used to edit UuT Lists.

5.1 The Test Bench Configuration File Editor

The Test Bench Configuration File Editor (TBCF Editor) is used to view and edit Test Bench Configuration Files (file extension "tbc"). These files are required by the Test Handler for text execution. For a description of the Test Bench Configuration File, refer to the section "The Test Bench Configuration File" on page 159.

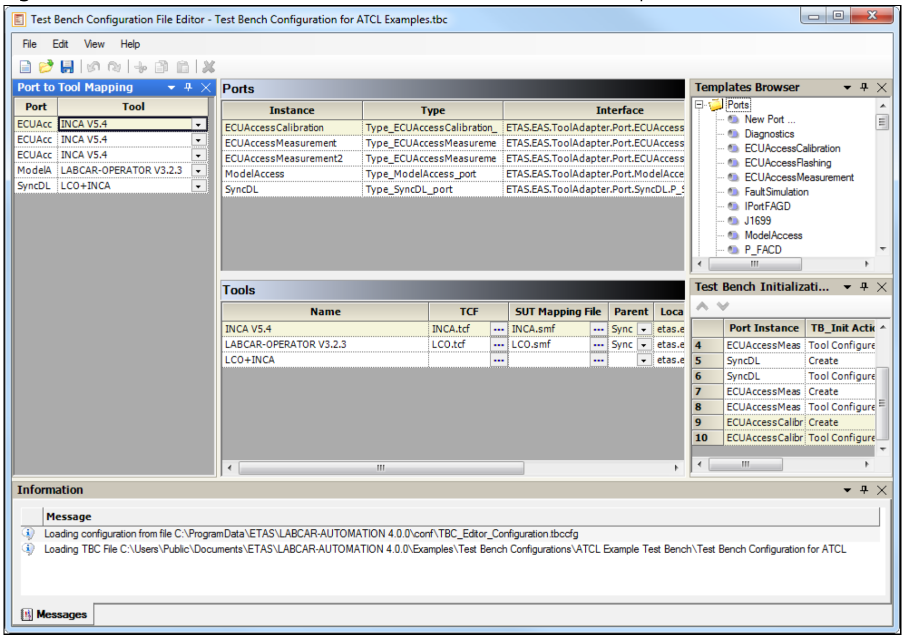
Only one Test Bench Configuration File (TBCF) can be open at a time in the editor – if several Test Bench Configuration Files are being edited, several editors can be opened.

To open the Test Bench Configuration File Editor

- In the Windows Start menu select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2.3** → **Editors** → **Test Bench Configuration File Editor**.

  The Test Bench Configuration File Editor opens.

Fig. 5-1 shows the window of the TBCF Editor with an open file.


Test Bench Configuration File Editor - Test Bench Configuration for ATCL Examples.tbc

File Edit View Help

Port to Tool Mapping

Port	Tool
ECUAcc	INCA V5.4
ECUAcc	INCA V5.4
ECUAcc	INCA V5.4
ModelA	LABCAR-OPERATOR V3.2.3
SyncDL	LCO+INCA

Ports

Instance	Type	Interface
ECUAccessCalibration	Type_ECUAccessCalibration_	ETAS.EAS.ToolAdapter.Port.ECUAccess
ECUAccessMeasurement	Type_ECUAccessMeasureme	ETAS.EAS.ToolAdapter.Port.ECUAccess
ECUAccessMeasurement2	Type_ECUAccessMeasureme	ETAS.EAS.ToolAdapter.Port.ECUAccess
ModelAccess	Type_ModelAccess_port	ETAS.EAS.ToolAdapter.Port.ModelAcce
SyncDL	Type_SyncDL_port	ETAS.EAS.ToolAdapter.Port.SyncDL.P_

Tools

Name	TCF	SUT Mapping File	Parent	Loca
INCA V5.4	INCA.tcf	INCA.smf	Sync	etas.e
LABCAR-OPERATOR V3.2.3	LCO.tcf	LCO.smf	Sync	etas.e
LCO+INCA				etas.e

Templates Browser

- Ports
  - New Port ...
  - Diagnostics
  - ECUAccessCalibration
  - ECUAccessFlashing
  - ECUAccessMeasurement
  - FaultSimulation
  - IPortFAGD
  - J1699
  - ModelAccess
  - P_FACD

Test Bench Initializati...

	Port Instance	TB_Init Acti
4	ECUAccessMeas	Tool Configure
5	SyncDL	Create
6	SyncDL	Tool Configure
7	ECUAccessMeas	Create
8	ECUAccessMeas	Tool Configure
9	ECUAccessCalibr	Create
10	ECUAccessCalibr	Tool Configure

Information

```
Message
Loading configuration from file C:\ProgramData\ETAS\LABCAR-AUTOMATION 4.0.0\conf\TBC_Editor_Configuration.tbccfg
Loading TBC File C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Bench Configurations\ATCL Example Test Bench\Test Bench Configuration for ATCL
```

Messages

Fig. 5-1 The Test Bench Configuration File Editor User Interface

The user interface of the TBCF Editor consists of the following parts:

- Ports

  All ports of the current Test Bench Configuration File are displayed under "Ports". Each row contains one port with its attributes (apart from "toolRef").

  For a precise description of the port list and the user actions in this list refer to the section "Ports" on page 149.

- Tools

  All tools of the Test Bench Configuration File are displayed under "Tools". Each row contains a tool with its attributes (apart from "instance").

  For a precise description of the tool list and the user actions in this list refer to the section "Tools" on page 152.
- The "Port to Tool Mapping" Window

  This is where the assignment between ports and tools is finally defined.

  For a precise description of this window and the user actions in this window refer to the section "The "Port to Tool Mapping" Window" on page 154.
- The "Test Bench Initialization Order" Window

  The order of the TB_Init steps is displayed and edited here. The first column contains the name of the relevant port instance, the second column the relevant TB_Init step ("Create" or "ConfigureTool").

  For a precise description of this window and the user actions in this window refer to the section "The "Test Bench Initialization Order" Window" on page 155.
- Template Browser

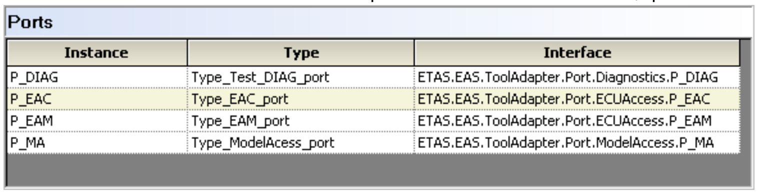
  The Template Browser contains a range of predefined tools and ports. The user can use these templates in the current TBC File. It is also possible to create new ports and tools and save these as templates.

  A precise description of the Template Browser and the use actions in it can be found in the section "The Template Browser" on page 157.
- The "Information" Window

  This is the window familiar from all LABCAR-AUTOMATION applications in which information and error messages are issued.

5.1.1 Ports

All ports of the current Test Bench Configuration File are displayed under "Ports". Each row contains one port with its attributes (apart from "toolRef").



Ports

Instance	Type	Interface
P_DIAG	Type_Test_DIAG_port	ETAS.EAS.ToolAdapter.Port.Diagnostics.P_DIAG
P_EAC	Type_EAC_port	ETAS.EAS.ToolAdapter.Port.ECUAccess.P_EAC
P_EAM	Type_EAM_port	ETAS.EAS.ToolAdapter.Port.ECUAccess.P_EAM
P_MA	Type_ModelAcess_port	ETAS.EAS.ToolAdapter.Port.ModelAccess.P_MA

Fig. 5-2 The List of Ports

The following table describes the purpose of the ports:

Name of Port	Purpose	Example Tool
P_DIAG	Access to diagnostic tool	ODX-LINK
P_EAC	Calibration access to ECU	INCA
P_EAF	Access to flash tool	INCA/ODX-FLASH
P_EAM	Measure access to ECU	INCA
P_FACD	File access (hierarchical data)	

Name of Port	Purpose	Example Tool
P_FATD	File access (tabular data)	
P_FS	Access to failure simulation	LABCAR-PINCONTROL
P_MA	Access to the simulation model (DVE)	LABCAR-OPERATOR
P_SyncDL	Access to synchronous data logging of INCA or LABCAR-OPERATOR	INCA/LABCAR-OPERATOR

The meaning of these attributes is described in the section "Ports" on page 160.

Note

As every port has to be assigned to a tool, ports can only be added if at least one tool has been defined.

To add a port

- Select **Add Port** from the shortcut menu of the list of ports.

  If ports are already available in the Template Browser, the submenu contains these for selection purposes.

or

- Select a port from the Template Browser (see section 5.1.5 on page 157) and drag it to the list of ports keeping the mouse button pressed down.

or

- Select a port from the Template Browser (see section 5.1.5 on page 157).
- Select **Add to TBC File** from the shortcut menu of the Template Browser.

  The port is added. If a port with the same name already exists, the name of the added port is extended by a number.

Note

The name of a port instance has to be unique within a TBCF!

To export a port as a template

- Select the port you want to export from the list of ports.
- Select **Export to Templates** from the shortcut menu.

  If the selected port is not available in the Template Browser, it is added to it. (see "The Template Browser" on page 157).

To cut a port

- Select the port you want to cut from the list of ports.
- Select **Cut** from the shortcut menu.

  The port is removed from the list and saved in the clipboard.

To copy a port

- Select the port you want to copy from the list of ports.
- Select **Copy** from the shortcut menu.

  A copy of the port is saved in the clipboard.

To insert a port

- Select **Paste** from the shortcut menu.

  The port in the clipboard is added.

To delete a port

- Select the port you want to delete.
- Select **Delete** from the shortcut menu.

  The port is deleted from the list.

To set the optimal column width

- Select **Optimal Column Width** from the shortcut menu.

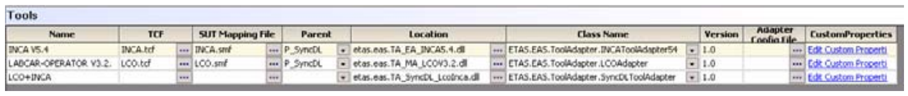
  The table column widths are changed so that the information contained in the columns is displayed in the best possible way.

5.1.2 Tools

All tools of the Test Bench Configuration File are displayed under ”Tools”. Each row contains a tool with its attributes (apart from ”instance”).

The meaning of these attributes is described in the section "Tools" on page 161.

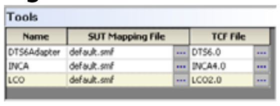
As the scope of the information displayed is relatively extensive (see Fig. 5-3), the number of columns can be limited to the three attributes ”Name”, ”SUT Mapping File” and ”TCF File” (see Fig. 5-4).



Tools

Name	TCF	SUT Mapping File	Parent	Location	Class Name	Version	Adapter Config File	CustomProperties
INCA V5.4	INCA.tcf	INCA.smf	P_SyncDL	etas.eas.TA_EA_INCA5.4.dll	ETAS.EAS.ToolAdapter.INCAToolAdapter54	1.0		Edit Custom Properti
LABCAR-OPERATOR V3.2.	LCO.tcf	LCO.smf	P_SyncDL	etas.eas.TA_MA_LCOV3.2.dll	ETAS.EAS.ToolAdapter.LCOAdapter	1.0		Edit Custom Properti
LCO+INCA				etas.eas.TA_SyncDL_LcoInca.dll	ETAS.EAS.ToolAdapter.SyncDLToolAdapter	1.0		Edit Custom Properti

Fig. 5-3 The List of Tools (”Extended View”)



Tools

Name	SUT Mapping File	TCF File
DTS6Adapter	default.smf	DTS6.0
INCA	default.smf	INCA4.0
LCO	default.smf	LCO2.0

Fig. 5-4 The List of Tools (”Basic View”)

To set the scope of information displayed

- Select **Show Basic View** or **Show Extended View** from the shortcut menu of the ”Tools” table.

  The attributes are set according to the scope selected.

To add a tool

- Select **Add Tool >>** from the shortcut menu of the list of tools.

  If tools are already available in the Template Browser, the submenu contains these for selection purposes.

  Otherwise, you can add a new tool by selecting **New Tool**.

or

- Select a tool from the Template Browser (see section 5.1.5 on page 157) and drag it to the list of tools keeping the mouse button pressed down.

or

- Select a tool from the Template Browser (see section 5.1.5 on page 157).

- Select **Add to TBC File** from the shortcut menu of the Template Browser.

  The tool is added. If a tool with the same name already exists, the name of the tool added is extended by a number.

Note

The name of a tool instance has to be unique within a TBCF!

To select an SUT Mapping File or a Tool Configuration File for the relevant tool, proceed as follows:

To select further configuration files

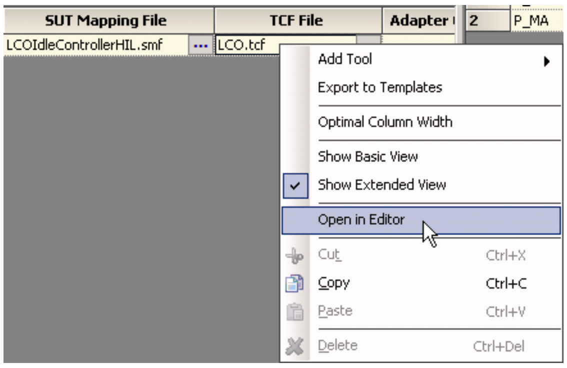
- Click the "..." icon in the "SUT Mapping File" or "TCF" column.

  A file selector window opens.
- Select the required file and click **OK**.

To open a selected configuration file in the editor, proceed as follows:

To open configuration files in the editor

- Right-click the table cell containing the file to be opened.
- Select **Open in Editor** from the shortcut menu.



The file is opened in the corresponding editor.

To export a tool as a template

- Select the tool you want to export from the list of tools.
- Select **Export to Templates** from the shortcut menu.

  If the selected tool is not available in the Template Browser, it is added to it. (see "The Template Browser" on page 157).

To cut a tool

- Select the tool you want to cut from the list of tools.
- Select **Cut** from the shortcut menu.

  The tool is removed from the list and saved in the clipboard.

To copy a tool

- Select the tool you want to copy from the list of tools.
- Select **Copy** from the shortcut menu.

  A copy of the tool is saved in the clipboard.

To add a tool

- Select **Paste** from the shortcut menu.

  The tool in the clipboard is added.

To delete a tool

- Select the tool you want to delete from the list of tools.
- Select **Delete** from the shortcut menu.

  The tool is deleted from the list.

Note

If the selected tool is referenced by a port, it cannot be deleted!

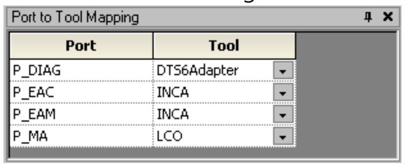
To set the optimal column width

- Select **Optimal Column Width** from the shortcut menu.

  The table column widths are changed so that the information contained in the columns is displayed in the best possible way.

5.1.3 The "Port to Tool Mapping" Window

This is where the assignment between ports and tools is finally defined.



Port to Tool Mapping

Port	Tool
P_DIAG	DTS6Adapter
P_EAC	INCA
P_EAM	INCA
P_MA	LCO

Fig. 5-5 Port to Tool Mapping

The window consists of two columns: The first column lists all ports contained in the TBCF; in the second column, the relevant tools can be assigned to these ports.

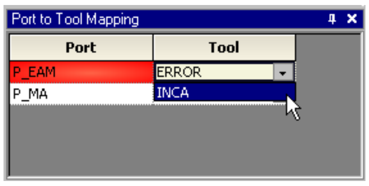
Note

A tool has to be assigned to every port listed in the Test Bench Configuration File!

If a new port is added to the list of ports, it also appears in the list of the "Port to Tool Mapping" window. You now have to assign it the relevant tool.

To assign a tool to a port

- Select the port (on a red background) which as yet has no tool assigned to it from the list.
- Click the selection symbol in the "Tool" column.
- Select the tool you want to assign to this port.

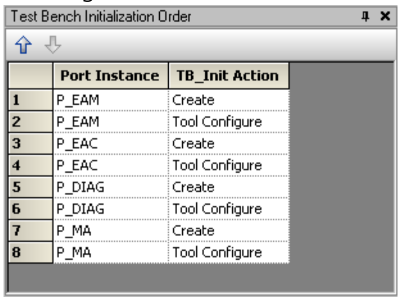


Port to Tool Mapping

Port	Tool
P_EAM	ERROR
P_MA	INCA

5.1.4 The "Test Bench Initialization Order" Window

In the "Test Bench Initialization Order" window, the order of the "Create" and "ConfigureTool" calls for TB_Init steps is determined.



Test Bench Initialization Order

	Port Instance	TB_Init Action
1	P_EAM	Create
2	P_EAM	Tool Configure
3	P_EAC	Create
4	P_EAC	Tool Configure
5	P_DIAG	Create
6	P_DIAG	Tool Configure
7	P_MA	Create
8	P_MA	Tool Configure

Fig. 5-6 Test Bench Initialization Order

If a new port is added to the list of ports, the two corresponding TB_Init steps ("Create" and "ConfigureTool") are added at the end of the list.

Order of TB_Init Steps

The order of actions can be changed using the arrow buttons. There is, however, the limitation that the TB_Init step "ConfigureTool" cannot be called before the "Create" step with a port.

To change the order of the ports

- To move a port upwards, click the **Move Up** icon

or

- Select **Move Up** from the shortcut menu.

- To move a port downwards, click the **Move Down** icon.

or

- Select **Move Down** from the shortcut menu.

To reset the order

- To reset the order of the port instances to an alpha-betical order, select **Reset TB_Init Order** from the shortcut menu.

5.1.5 The Template Browser

The Template Browser contains a range of ports and tools which were previously assigned there as templates (see "To export a port as a template" on page 151 and "To export a tool as a template" on page 153).

From here, they can then, for example, be inserted into the list of ports or the list of tools per drag & drop (see "To add a port" on page 150 or "To add a tool" on page 152).

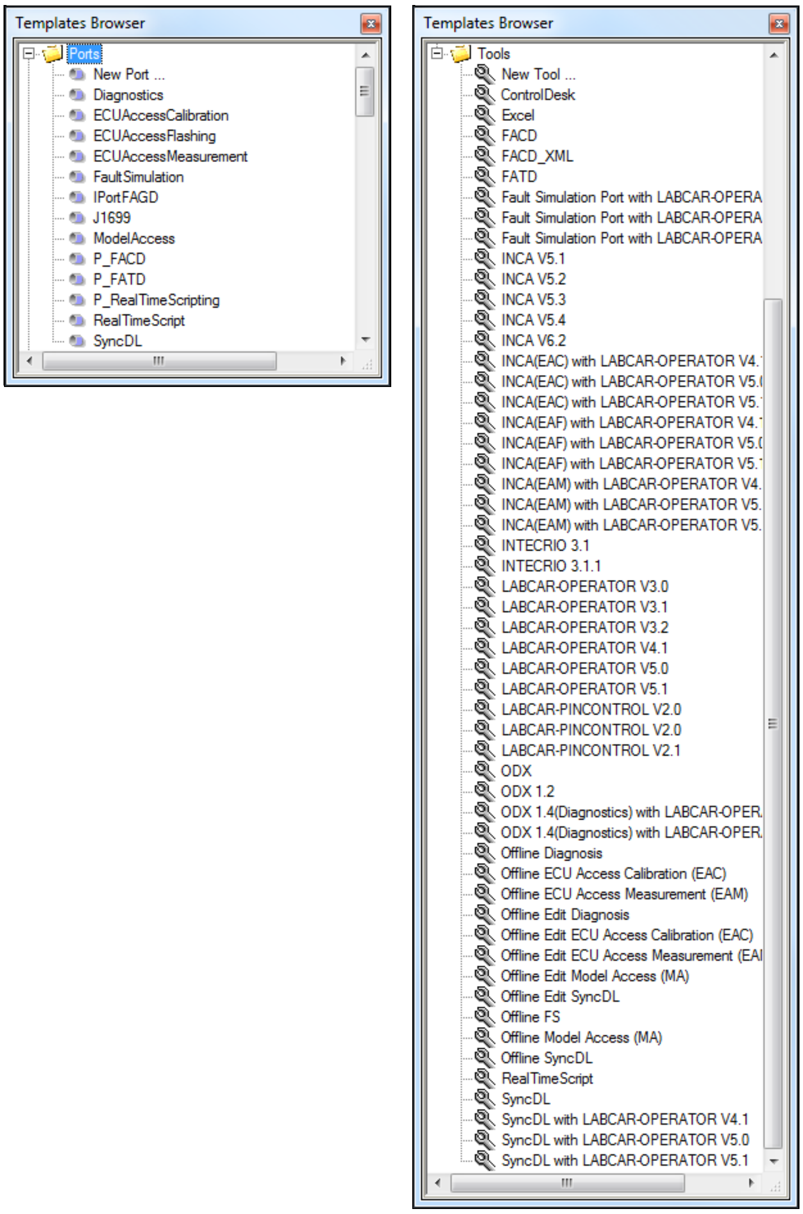

Templates Browser
Ports
New Port ...
Diagnostics
ECUAccessCalibration
ECUAccessFlashing
ECUAccessMeasurement
FaultSimulation
IPortFAGD
J1699
ModelAccess
P_FACD
P_FATD
P_RealTimeScripting
RealTimeScript
SyncDL
Templates Browser
Tools
New Tool ...
ControlDesk
Excel
FACD
FACD_XML
FATD
Fault Simulation Port with LABCAR-OPERA
Fault Simulation Port with LABCAR-OPERA
Fault Simulation Port with LABCAR-OPERA
INCA V5.1
INCA V5.2
INCA V5.3
INCA V5.4
INCA V6.2
INCA(EAC) with LABCAR-OPERATOR V4.
INCA(EAC) with LABCAR-OPERATOR V5.
INCA(EAC) with LABCAR-OPERATOR V5.
INCA(EAF) with LABCAR-OPERATOR V4.
INCA(EAF) with LABCAR-OPERATOR V5.
INCA(EAF) with LABCAR-OPERATOR V5.
INCA(EAM) with LABCAR-OPERATOR V4.
INCA(EAM) with LABCAR-OPERATOR V5.
INCA(EAM) with LABCAR-OPERATOR V5.
INTECRIO 3.1
INTECRIO 3.1.1
LABCAR-OPERATOR V3.0
LABCAR-OPERATOR V3.1
LABCAR-OPERATOR V3.2
LABCAR-OPERATOR V4.1
LABCAR-OPERATOR V5.0
LABCAR-OPERATOR V5.1
LABCAR-PINCONTROL V2.0
LABCAR-PINCONTROL V2.0
LABCAR-PINCONTROL V2.1
ODX
ODX 1.2
ODX 1.4(Diagnostics) with LABCAR-OPER
ODX 1.4(Diagnostics) with LABCAR-OPER
Offline Diagnosis
Offline ECU Access Calibration (EAC)
Offline ECU Access Measurement (EAM)
Offline Edit Diagnosis
Offline Edit ECU Access Calibration (EAC)
Offline Edit ECU Access Measurement (EA
Offline Edit Model Access (MA)
Offline Edit SyncDL
Offline FS
Offline Model Access (MA)
Offline SyncDL
RealTimeScript
SyncDL
SyncDL with LABCAR-OPERATOR V4.1
SyncDL with LABCAR-OPERATOR V5.0
SyncDL with LABCAR-OPERATOR V5.1

Fig. 5-7 The Template Browser – Ports and Tools

5.1.6 The Test Bench Configuration File Editor Main Menu

The Test Bench Configuration File Editor Main Menu contains the following items:

- **File**

  See "The ”File” Menu" on page 158.
- **Edit**

  See "The ”Edit” Menu" on page 158.
- **View**

  See "The ”View” Menu" on page 159.
- **Help**

  See "The ”Help” Menu" on page 159.

The ”File” Menu

- **File → New**

  Creates a new Test Bench Configuration File.
- **File → Open**

  Opens an existing Test Bench Configuration File.
- **File → Save**

  Saves the current file.
- **File → Save As**

  Saves the current file under a different file name.
- **File → Recent Files >>**

  Shows the list of the four files last opened from which one can be selected to be opened.
- **File → Exit**

  Exits the TBC Editor.

The ”Edit” Menu

- **Edit → Undo**

  Reverses the last step.
- **Edit → Redo**

  Redoes a step previously undone.
- **Edit → Cut**

  Cuts the selected port or tool and stores it in the clipboard.
- **Edit → Copy**

  Copies the selected port or tool to the clipboard.
- **Edit → Paste**

  Adds the port or tool in the clipboard to the current port list or tool list or Template Browser.

- **Edit → Delete**

  Deletes the selected port or tool.

The "View" Menu

- **View → Port to Tool Mapping**

  Opens/closes the "Port to Tool Window" window.
- **View → Test Bench Initialization**

  Opens/closes the "Test Bench Initialization Order" window.
- **View → Templates Browser**

  Opens/closes the Template Browser.
- **View → Information**

  Opens/closes the "Information" window.

The "Help" Menu

- **Help → User's Guide**

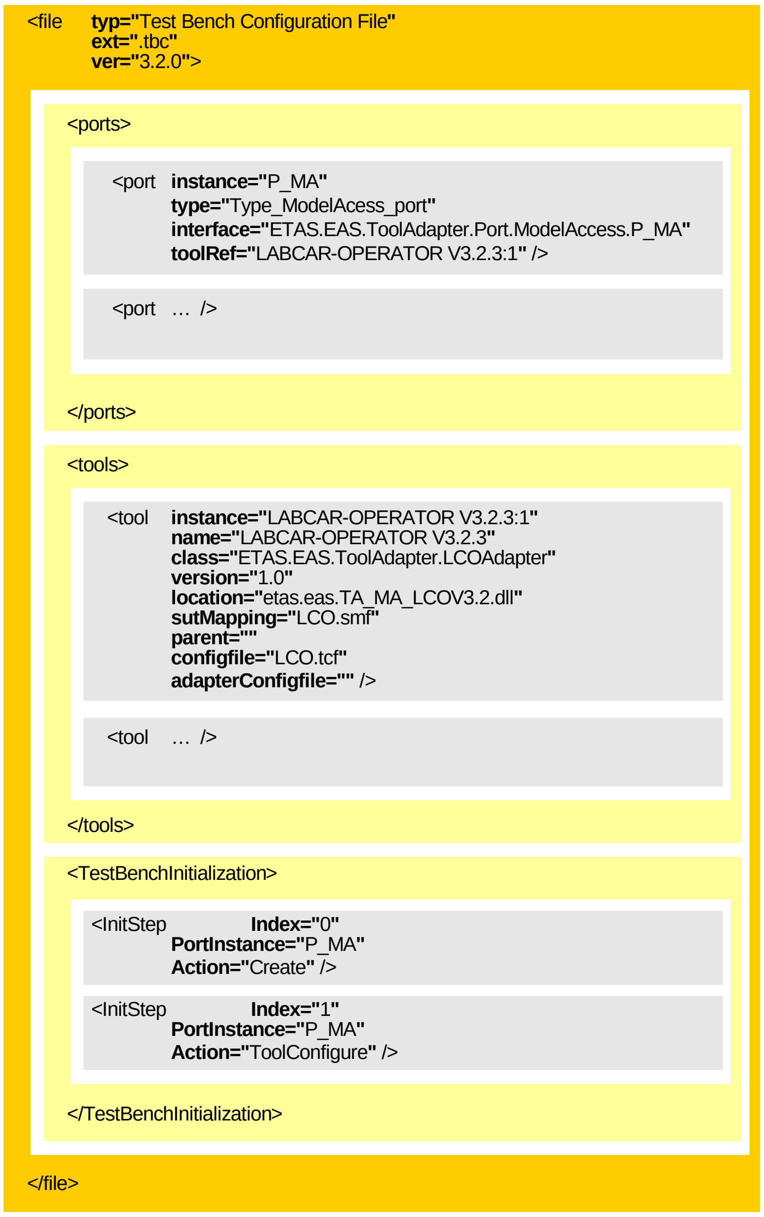
  Opens the English User's Guide (PDF).
- **Help → System Info**

  Opens an information window containing details of the current versions of the components of LABCAR-AUTOMATION 4.2.3.
- **Help → About**

  Opens a window containing general information.

5.1.7 The Test Bench Configuration File

A Test Bench Configuration File (TBCF) is an XML file which contains information about ports, tools and initialization sequences for the test bench. The main purpose of the Test Bench Configuration File (TBCF) is to map port instances used with the test cases to specific tool instances. The TBCF also provides unambiguous configuration for all tools of the current test environment.



```
<file    typ="Test Bench Configuration File"
         ext=".tbc"
         ver="3.2.0">

  <ports>

    <port  instance="P_MA"
           type="Type_ModelAcess_port"
           interface="ETAS.EAS.ToolAdapter.Port.ModelAccess.P_MA"
           toolRef="LABCAR-OPERATOR V3.2.3:1" />

    <port  … />

  </ports>

  <tools>

    <tool  instance="LABCAR-OPERATOR V3.2.3:1"
           name="LABCAR-OPERATOR V3.2.3"
           class="ETAS.EAS.ToolAdapter.LCOAdapter"
           version="1.0"
           location="etas.eas.TA_MA_LCOV3.2.dll"
           sutMapping="LCO.smf"
           parent=""
           configfile="LCO.tcf"
           adapterConfigfile="" />

    <tool  … />

  </tools>

  <TestBenchInitialization>

    <InitStep        Index="0"
           PortInstance="P_MA"
           Action="Create" />

    <InitStep        Index="1"
           PortInstance="P_MA"
           Action="ToolConfigure" />

  </TestBenchInitialization>

</file>
```

Fig. 5-8 The Structure of the XML File for the Test Bench Configuration

The TBCF contains the following information (see Fig. 5-8):

Ports

Every port used in the test cases has to be declared in the TBCF.

The XML element "port" has the following attributes:

- instance=

  Name of the port instance.

- type=

  The definition of port types for all port instances (TTCN3 name).

- interface=

  The description of the interface for each port instance.
- toolRef=

  The definition of a tool instance for the respective port instance. This tool instance also has to be defined in the TBCF (see below).

Tools

Every port references a tool instance – this is described via the attributes of the XML element ”tool”.

- instance=

  The definition of the instance name – the tool instance is referenced by the port instances described above under this name.
- name=

  Definition of a name for better readability.
- class=

  Definition of the tool adapter class to be used.
- version=

  Version of the tool adapter class.
- location=

  Name of the tool adapter DLL.
- sutMapping=

  Definition of the SUT Mapping File.
- parent=

  Optional definition of a parent-tool instance to which the current tool instance is related.
- configfile=

  Definition of a Tool Configuration File (TCF).
- adapterConfigfile=

  Definition of special adapter configuration files (optional). If the tool adapter needs additional special configuration files, these can be declared here.

TestBenchInitialization

In the ”TestBenchInitialization” element, all steps for test bench initialization (”InitStep” element) are declared.

The XML element ”InitStep” has the following attributes:

- Index=

  Order of the TB_Init steps - the enumeration must start with 0 and must not contain any gaps.
- PortInstance=

  Name of the port instance to which the relevant TB_Init step belongs.

- Action=

  The actual TB_Init action. Every port requires a ”Create” and a ”ConfigureTool” entry.

Note

The TB_Init step ”ConfigureTool” must not be called before the ”Create” step! However, several ”Create” calls may be executed before a ”ConfigureTool” call.

5.2 The Tool Configuration File Editor

The TCF Editor is used to view and edit a Tool Configuration File (extension "tcf").

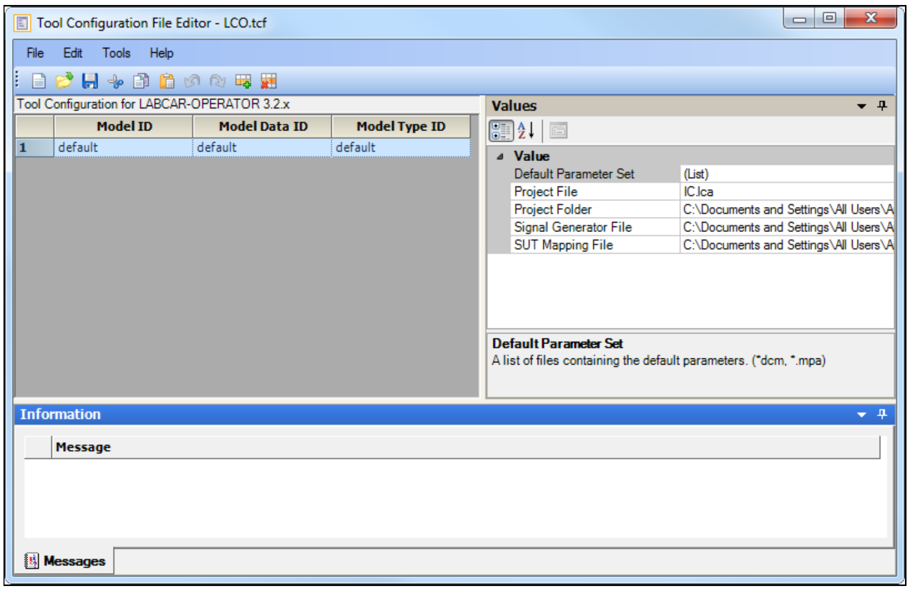
For a description of the Tool Configuration File, refer to the section "The Tool Configuration File" on page 167.

One Tool Configuration File can be opened in the editor – if several files are to be edited, several editors can be opened.

To open a Tool Configuration File

- Select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2** → **Editors** → **Tool Configuration File Editor**.
- The TCF Editor opens.
- Select **File** → **Open**.
- Select the Tool Configuration File (*.tcf) to be opened and click **OK**.

  The file is opened in the TCF Editor.



Tool Configuration for LABCAR-OPERATOR 3.2.x

	Model ID	Model Data ID	Model Type ID
1	default	default	default

Values

Value	
Default Parameter Set	(List)
Project File	IC.lca
Project Folder	C:\Documents and Settings\All Users\A
Signal Generator File	C:\Documents and Settings\All Users\A
SUT Mapping File	C:\Documents and Settings\All Users\A

Default Parameter Set
A list of files containing the default parameters. (*dcm, *.mpa)

	Message

The user interface of the TCF Editor consists of two parts:

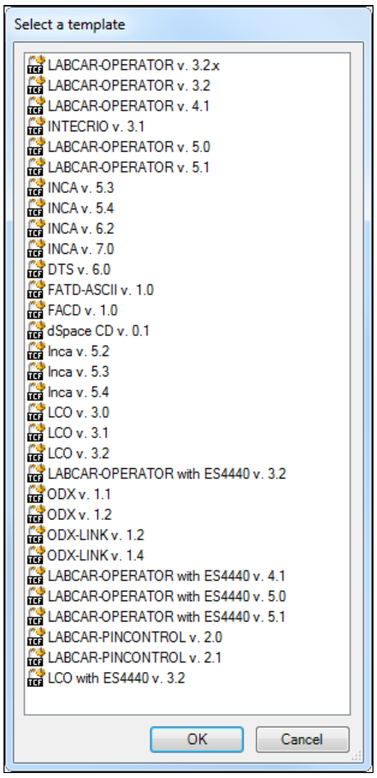
- a table with all keys of a configuration. Each line represents one configuration.
- the "Values" window in which the relevant values can be changed.

All names of the keys and values can be edited in the editor. It is also possible to copy names to the clipboard and then reinsert them. When reinserting, which key belongs to which name is observed – it can only be reinserted as belonging to this key. Identical names are avoided by adding a number.

To create a tool configuration

- To create a new Tool Configuration File, select **File** → **New**.

  A window opens from which you can select a template.



Select a template

LABCAR-OPERATOR v. 3.2x
LABCAR-OPERATOR v. 3.2
LABCAR-OPERATOR v. 4.1
INTECRIO v. 3.1
LABCAR-OPERATOR v. 5.0
LABCAR-OPERATOR v. 5.1
INCA v. 5.3
INCA v. 5.4
INCA v. 6.2
INCA v. 7.0
DTS v. 6.0
FATD-ASCII v. 1.0
FACD v. 1.0
dSpace CD v. 0.1
Inca v. 5.2
Inca v. 5.3
Inca v. 5.4
LCO v. 3.0
LCO v. 3.1
LCO v. 3.2
LABCAR-OPERATOR with ES4440 v. 3.2
ODX v. 1.1
ODX v. 1.2
ODX-LINK v. 1.2
ODX-LINK v. 1.4
LABCAR-OPERATOR with ES4440 v. 4.1
LABCAR-OPERATOR with ES4440 v. 5.0
LABCAR-OPERATOR with ES4440 v. 5.1
LABCAR-PINCONTROL v. 2.0
LABCAR-PINCONTROL v. 2.1
LCO with ES4440 v. 3.2

- Select the tool and click **OK**.

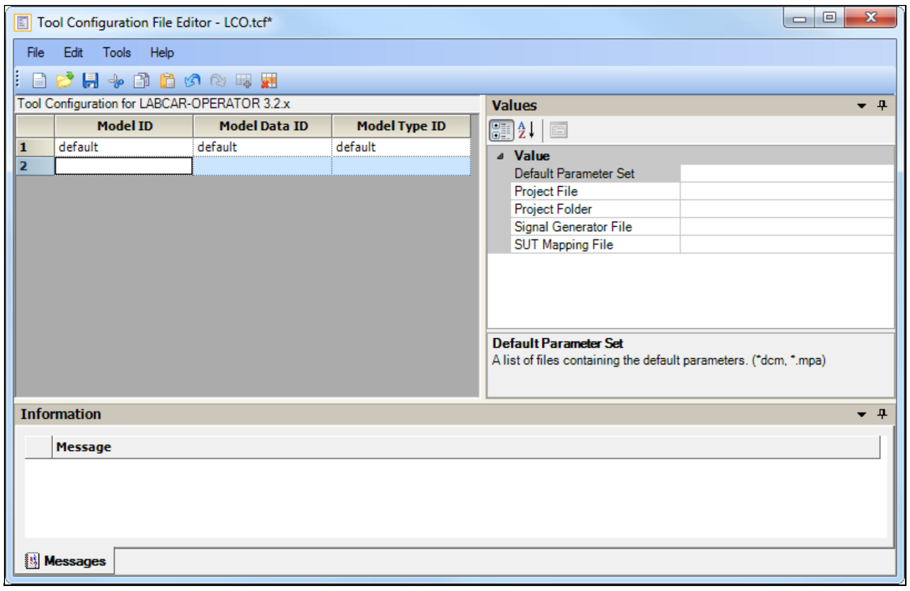
  The file is created.

- To add a tool configuration, select **Edit** → **Add**.

or

- Click the **Add** icon in the toolbar.

  A new row is added to the list of tool configurations.



Tool Configuration for LABCAR-OPERATOR 3.2.x

	Model ID	Model Data ID	Model Type ID
1	default	default	default
2			

Value	
Default Parameter Set	
Project File	
Project Folder	
Signal Generator File	
SUT Mapping File	

Default Parameter Set
A list of files containing the default parameters. (*dcm, *.mpa)

Message

- Enter the relevant keys.
- To fill the list of values, click the relevant entry.
- Click the **...** button to select the file required.
- Select **File** → **Save**.

To remove a tool configuration

- Select the configuration to be deleted.
- Select **Edit** → **Delete**.

or

- Click the **Delete** icon in the toolbar.

  The selected configuration is removed.

5.2.1 Parsing the TargetServer Log File

To read a flash configuration, for example, it is useful to take a look at the target server log file. For this purpose, there is a parser which is launched using **Tools** → **Parse TargetServer Log File**.

5.2.2 Configuration of the TCF Editor

To keep the TCF Editor open for non-ETAS tools, a configuration file is supplied which can be extended by the user. This means that the table of keys and the values in the "Values" field can be adapted for the new tool.

The name of this file is `TCF_Editor_Configuration.xml` and it is located in the directory
`<drive>:\Program Files (x86)\ETAS\`
`LABCAR-AUTOMATION 4.2\TestTools\Bin\`.

5.2.3 Offline Tool Adapter

An offline tool adapter makes it possible for you to work with a project without the corresponding tool being available.

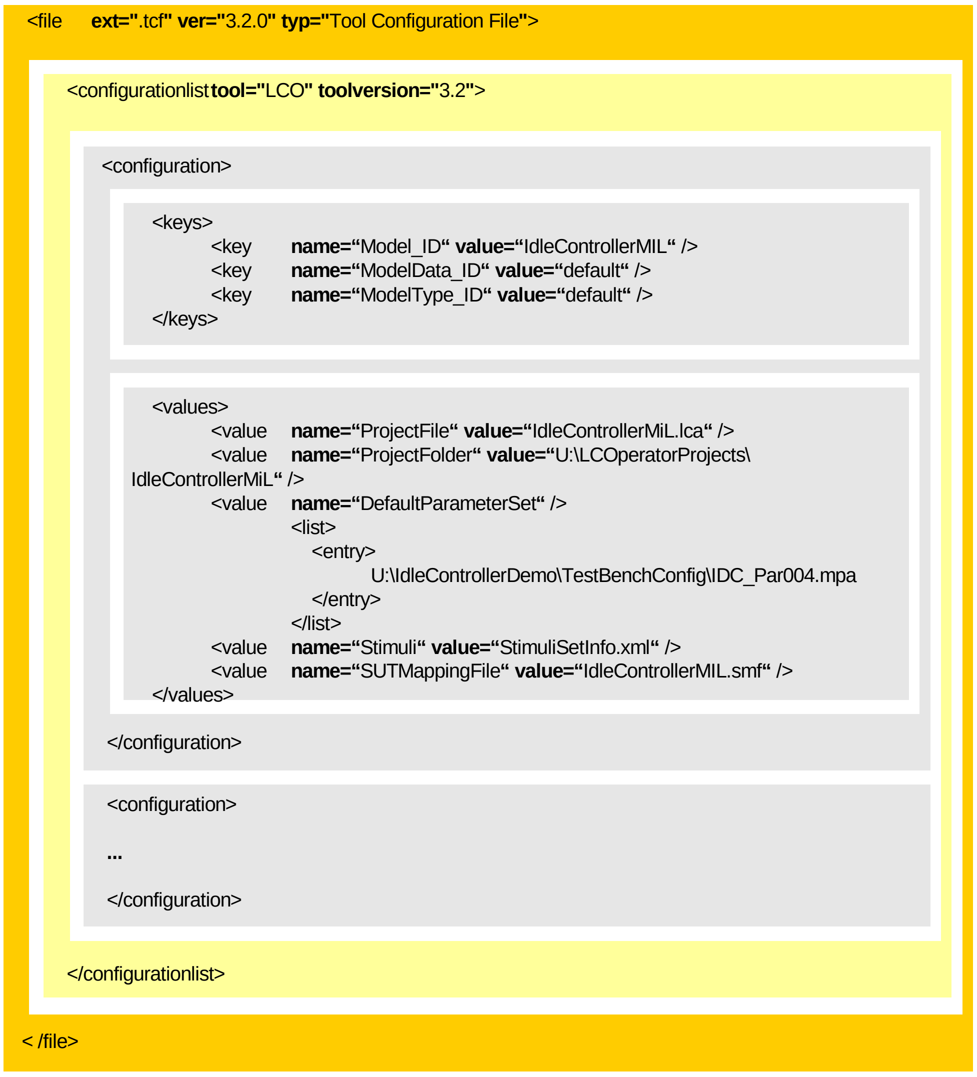
This tool adapter simulates the ”responses” of the real tool – these are stored in a special tool configuration file which is declared in the Test Bench Configuration File in the XML element <Tools> in the attribute ”adapterConfigfile” (see "adapterConfigfile=" on page 161).

One of the folders that also contains the sample files for the tutorial contains an example of this kind of configuration file:

```
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION
4.2\Examples\Test Bench Configurations\Demo Test
Bench\Offline\Offline.txt
```

5.2.4 The Tool Configuration File

A Tool Configuration File (TCF) contains valid configurations for a tool of the test bench which is required to run the test project. A separate Tool Configuration File has to be available for every tool.



```
<file    ext=".tcf" ver="3.2.0" typ="Tool Configuration File">

  <configurationlist tool="LCO" toolversion="3.2">

    <configuration>

      <keys>
              <key      name=“Model_ID“ value=“IdleControllerMIL“ />
              <key      name=“ModelData_ID“ value=“default“ />
              <key      name=“ModelType_ID“ value=“default“ />
      </keys>

      <values>
              <value    name=“ProjectFile“ value=“IdleControllerMiL.lca“ />
              <value    name=“ProjectFolder“ value=“U:\LCOperatorProjects\
    IdleControllerMiL“ />
              <value    name=“DefaultParameterSet“ />
                        <list>
                           <entry>
                                  U:\IdleControllerDemo\TestBenchConfig\IDC_Par004.mpa
                           </entry>
                        </list>
              <value    name=“Stimuli“ value=“StimuliSetInfo.xml“ />
              <value    name=“SUTMappingFile“ value=“IdleControllerMIL.smf“ />
      </values>

    </configuration>

    <configuration>

    ...

    </configuration>

  </configurationlist>

< /file>
```

Fig. 5-9 The Structure of the XML File for the Tool Configuration

A tool configuration consists of a range of key/value pairs resulting in an unambiguous configuration of a tool. These keys and how many keys there are depend on the type of port used.

In the following tables, you will find the keys and values of all tools supported by LABCAR-AUTOMATION 4.2.3.

Note

A Tool Configuration File (`*.tcf`) always contains the configuration for one tool only!

LABCAR-OPERATOR

Key <Name (XML Attribute)>
Model ID <Model_ID>
Model Data ID <ModelData_ID>
Model Type ID <ModeType_ID>

Value <Name (XML Attribute)>	Type	Comment
Default Parameter Set <DefaultParameterSet>	File list	List of parameter sets (*.dcm, *.mpa files)
Project File <ProjectFile> *	File	The LABCAR-OPERATOR project file (*.lca)
Project Folder <ProjectFolder> *	Folder	Directory where the project file is stored
Stimuli <Stimuli>	File	Reference to the configuration file for the Signal Generator (optional)
SUT Mapping File <SUTMappingFile>	File	SUT Mapping File (optional) - if a file is specified here, this file is used instead of the one specified in the Test Bench Configuration File.
Experiment <Experiment> **	File	The file for the LABCAR-OPERATOR experiment (*.eex)
Workspace <Workspace> **	File	The file that contains the workspace (LABCAR-EE) (*.eew)
* Only for LABCAR-OPERATOR V2.0 and V3.x ** Only for LABCAR-OPERATOR V4.x		

LABCAR-OPERATOR with ES4440

If an ES4440 Compact Failure Simulation Module is used to simulate errors, the list of values is extended with the following:

Value <Name (XML Attribute)>	**Type**	**Comment**
ES4440 - Cleanup Relay <ES4440_CleanupRelay>	String	Runs a clean-up of the relays of the specified channels
ES4440 - ETH Card <ES4440_EthCard>	String	The Ethernet card of the real-time PC to be used for communicating with the ES4440
ES4440 - ETH Interface <ES4440_EthInterface>	String	Communication interface: 0 = Ethernet (realtime) 1 = Ethernet (non-realtime) 2 = CAN (IXXAT CAN Board)
ES4440 - IP Master <ES4440_IP_Master>	String	The IP address of the master
ES4440 - LoadFlag <ES4440_Load>	String	Setting the Load flag in ES4440 com-mands: "with" or "without"
ES4440 - Number of devices <ES4440_numberOfDevices >	String	The number of ES4440 modules
ES4440 - Potential mapping HC: GND <ES4440_HC_UBat>	String	Assignment of "GND" to one of the ES4440 potentials (+UBatt_A, –UBatt_A, +UBatt_B, –UBatt_B) for high-current channels
ES4440 - Potential mapping HC: UBat <ES4440_HC_UBat>	String	Assignment of "UBat" to one of the ES4440 potentials (+UBatt_A, –UBatt_A, +UBatt_B, –UBatt_B) for high-current channels
ES4440 - Potential mapping HC: UPot <ES4440_HC_UBat>	String	Assignment of "UPotential" to one of the ES4440 potentials (+UBatt_A, –UBatt_A, +UBatt_B, –UBatt_B) for high-current channels
ES4440 - Potential mapping HV: GND <ES4440_HV_UBat>	String	Assignment of "GND" to one of the ES4440 potentials (+UBatt_C, –UBatt_C) for high-voltage channels
ES4440 - Potential mapping HV: UBat <ES4440_HV_UBat>	String	Assignment of "UBat" to one of the ES4440 potentials (+UBatt_C, –UBatt_C) for high-voltage channels
ES4440 - Potential mapping HV: UPot <ES4440_HV_UBat>	String	Assignment of "UPotential" to one of the ES4440 potentials (+UBatt_C, –UBatt_C) for high-voltage channels
ES4440 - Test Fuses <ES4440_TestFuses>	String	Fuse test for ES4440 modules ("true" or "false")
ES4440 - Wire Harness File <ES4440_WireHarnessFile>	File	Description of the wire harness

LABCAR-PINCONTROL V2.0

The Tool Configuration File for LABCAR-PINCONTROL V2.0 contains a subset of the values of the file described above for LABCAR-OPERATOR with ES4440.

INCA

Key <Name (XML-Attribute)>
EA ID <EA_ID>
Device ID <DeviceID>
Hardware Layer ID <HWLayerID>
Protocol ID <ProtocolID>

Value <Name (XML Attribute)>	Type	Comment
ASAP File <ASAPFile>	File	Name of the ASAP file which is to be used for the INCA configuration
Database Export File <DBEx-portFile>	File	Name of the INCA database export file which is required for the configuration of INCA
Device <Device>	String	Name of the INCA device
ECU value get Timeout <ECUValueGetTimeout>	String	Timeout value for ECU access
Experiment <Experiment>	String	Name of the experiment
Experiment Folder <ExperimentFolder>	String	Name of the experiment folder
HEX File <HEXFile>	File	Name of the HEX file which is to be used for the INCA configuration
IncaDB <IncaDB>	Folder	Directory containing the INCA data-base
SUT Mapping File <SUTMappingFile>	File	SUT Mapping File (optional)
Work Base <WorkBase>	String	Name of the INCA workbase which is required for the configuration of INCA.
Workspace <Workspace>	String	Name of the workspace folder
Workspace Folder <WorkspaceFolder>	String	Name of the workspace
Flash Port <FlashPort>	String	The hardware port used for flashing
PROF Configuration File <PROFConfigFile>	File	Configuration file required by PROF
PROF Flash Parameter <FlashParameters>	String	Flash parameter required by PROF
Code/Data Files for flashing <CodeDataFiles>	File list	Additional code/data for flashing

ODX-LINK

Key <Name (XML-Attribute)>
Customer ID <Customer_ID>
ECU ID <ECU_ID>
Node ID <Node_ID>
Protocol ID <Protocol_ID>

Value <Name (XML Attribute)>	Type	Comment
Location <Location>	String	Location (ECU, Protocol Services etc.)
Project <Project>	String	Name of ODX project (without file extension)
SUT Mapping File <SUTMappingFile>	File	SUT Mapping File (optional) - if a file is specified here, it is used instead of the file specified in the Test Bench Configuration File.
Tool Configuration File <ToolConfigFile>	File	Not used
Vehicle Information Table <VehicleInformationTable>	String	Vehicle information from the diagnostic database

FATD1.0

Key <Name (XML-Attribut)>
UuT File ID <UuTFile_ID>
ASCII File ID <File_ID>

Value <Name (XML Attribute)>	Type	Comment
File Location <FileLocation>	File	Path and name of file
File Type <FileType>	String	Always ASCII
Row Separator Char <RowSeparatorChar>	ASCII character	Character for row separator
Column Separator Char <ColumnSeparatorChar>	ASCII character	Character for column separator

FACD1.0

Key <Name (XML-Attribut)>
UuT File ID <UuTFile_ID>
File ID <File_ID>

Value <Name (XML Attribute)>	Type	Comment
File Location <FileLocation>	File	Path and name of file
File Type <CDFileType>	String	Always XML
SUT Mapping File <SUTMappingFile>	File	Name of the SUT Mapping File

dSpace CD (ControlDesk)

Key <Name (XML Attribute)>
Model ID <Model_ID>
Model Data ID <ModelData_ID>
Model Type ID <ModeType_ID>

Value <Name (XML Attribute)>	**Type**	**Comment**
Platform Name <PlatformName>	String	Name of the platform to be activated
Project File <ProjectFile>	File	The ControlDesk project file)
Project Folder <ProjectFolder>	Folder	Directory where the project file is stored
Default Parameter Set <DefaultParameterSet>	File list	List of parameter sets (*.dcm, *.mpa files)
Stimuli <Stimuli>	File	Reference to the configuration file for the Signal Generator (optional)
SUT Mapping File <SUTMappingFile>	File	SUT Mapping File (optional) - if a file is specified here, this file is used instead of the one specified in the Test Bench Configuration File.

5.3 The Switch Definition File Editor

The Switch Definition File Editor is used to create and edit Switch Definition Files (*.lcasdt). These files are needed by the Test Case Developer and the Test Parameter Manager to prepare parameter objects of the type "Switch". For a description of the Switch Definition File, refer to the section "The Switch Definition File" on page 183.

Only one Switch Definition File can be open at a time in the editor – if several Switch Definition Files are being edited, several editors can be opened.

To launch the Switch Definition File Editor

- In the Windows Start menu select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2.3** → **Editors** → **Switch Definition File Editor**.

  The Switch Definition File Editor is launched and an empty Switch Definition File loaded.

To load a Switch Definition File

- Select **File** → **Open**.
- Select a file (with the file extension .lcasdt) from the file selector window.

  The selected file opens.

To create a new Switch Definition File

- If you want to create a new Switch Definition File, select **File** → **New**.

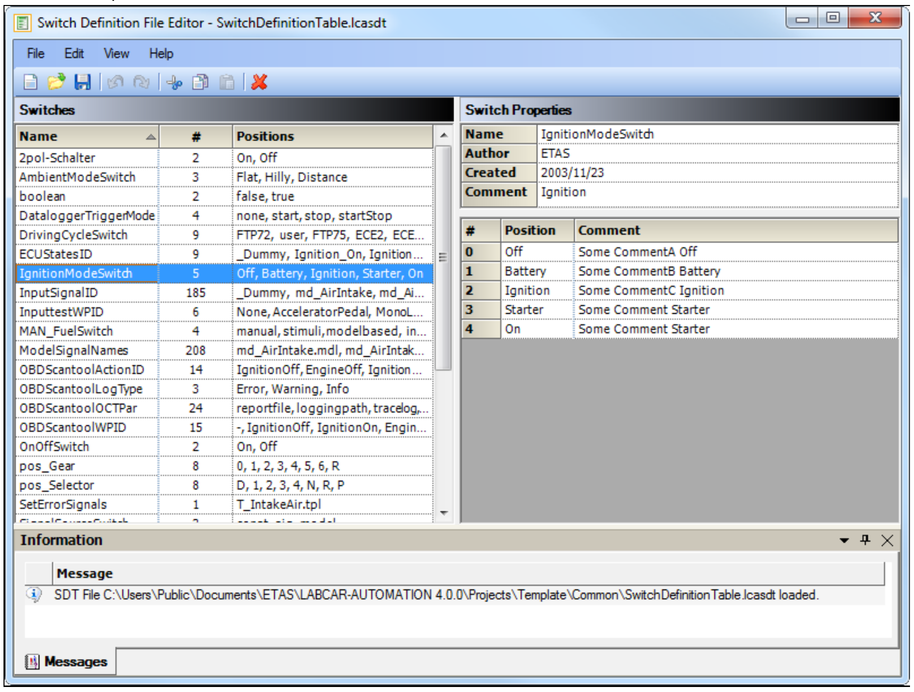
  An empty Switch Definition File is created.

  If a Switch Definition File with unsaved changes was open previously, you are prompted to save the changes if you require them.

To save a Switch Definition File

- To save an open Switch Definition File, select **File** → **Save**.
- If the file has not yet been saved or was saved under a different name, select **File** → **Save as**.

The following figure shows the user interface of the Switch Definition File Editor with an open file.


Switch Definition File Editor - SwitchDefinitionTable.lcasdt
File
Edit
View
Help
Switches
Switch Properties
Information
Messages

Name	#	Positions
2pol-Schalter	2	On, Off
AmbientModeSwitch	3	Flat, Hilly, Distance
boolean	2	false, true
DataloggerTriggerMode	4	none, start, stop, startStop
DrivingCycleSwitch	9	FTP72, user, FTP75, ECE2, ECE...
ECUStatesID	9	_Dummy, Ignition_On, Ignition...
IgnitionModeSwitch	5	Off, Battery, Ignition, Starter, On
InputSignalID	185	_Dummy, md_AirIntake, md_Ai...
InputtestWPID	6	None, AcceleratorPedal, MonoL...
MAN_FuelSwitch	4	manual, stimuli, modelbased, in...
ModelSignalNames	208	md_AirIntake.mdl, md_AirIntak...
OBDScantoolActionID	14	IgnitionOff, EngineOff, Ignition...
OBDScantoolLogType	3	Error, Warning, Info
OBDScantoolOCTPar	24	reportfile, loggingpath, tracelog,...
OBDScantoolWPID	15	-, IgnitionOff, IgnitionOn, Engin...
OnOffSwitch	2	On, Off
pos_Gear	8	0, 1, 2, 3, 4, 5, 6, R
pos_Selector	8	D, 1, 2, 3, 4, N, R, P
SetErrorSignals	1	T_IntakeAir.tpl

Name	IgnitionModeSwitch
Author	ETAS
Created	2003/11/23
Comment	Ignition

#	Position	Comment
0	Off	Some CommentA Off
1	Battery	Some CommentB Battery
2	Ignition	Some CommentC Ignition
3	Starter	Some Comment Starter
4	On	Some Comment Starter

Message
SDT File C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Projects\Template\Common\SwitchDefinitionTable.lcasdt loaded.

Fig. 5-10 Switch Definition File Editor with an Open File

The user interface consists of the following areas:

- Menu bar

  For a detailed description of the main menu of the Switch Definition File, refer to the section "The Main Menu of the Switch Definition File Editor" on page 180.

- Toolbar

  A description of the toolbar is contained in the section "The Toolbar" on page 182.

- The "Switches" Window

  This window displays all the switches defined in the Switch Definition File as well as the number and names of the switch positions.

  A description of this window is contained in the section "The "Switches" Window" on page 176.

- The "Switch Properties" Window

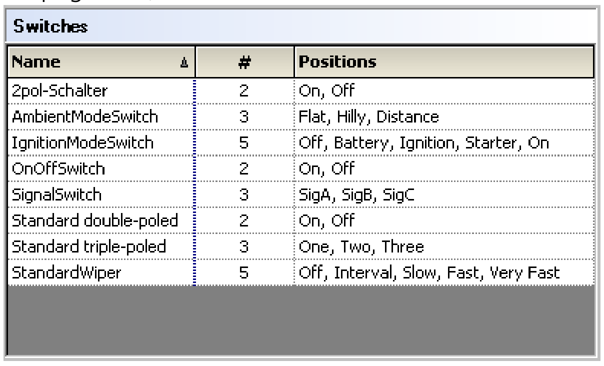
  This window displays all properties of a switch selected in the "Switches" window; the properties can also be edited here. A description of this window is contained in the section "The "Switch Properties" Window" on page 178.

- The "Information" Window

  This is the window familiar from all LABCAR-AUTOMATION applications in which information and error messages are issued.

5.3.1 The "Switches" Window

The "Switches" window displays all switches defined in the Switch Definition File with their name ("Name"), number of switch positions ("#") and their positions ("Positions") (see Fig. 5-13 on page 184, Tab. 5-2 on page 185 and Tab. 5-3 on page 185).



Switches

Name	#	Positions
2pol-Schalter	2	On, Off
AmbientModeSwitch	3	Flat, Hilly, Distance
IgnitionModeSwitch	5	Off, Battery, Ignition, Starter, On
OnOffSwitch	2	On, Off
SignalSwitch	3	SigA, SigB, SigC
Standard double-poled	2	On, Off
Standard triple-poled	3	One, Two, Three
StandardWiper	5	Off, Interval, Slow, Fast, Very Fast

Fig. 5-11 The "Switches" Window

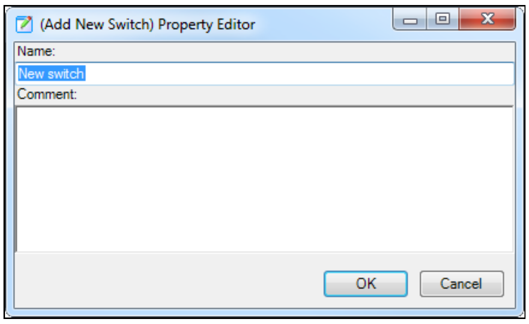
The entries in the list cannot be edited here – but you can add and delete switches.

All actions possible in this window can be accessed via the shortcut menu and are described below.

To create a new object of the type "Switch"

- Select **New Switch** from the shortcut menu.

  The "Add New Switch" window opens.



(Add New Switch) Property Editor

Name: New switch

Comment:

- Enter a name and possibly a comment.
- Click **OK**.

  The new switch (with a default position) is created.

  For details of how to create additional switch positions, refer to the section "The "Switch Properties" Window" on page 178).

To rename a switch

- Select the switch you want to rename.
- Select **Rename** from the shortcut menu.

  The "Rename Switch" window opens.

- Change the name and, if necessary, the comment.
- Click **OK**.

  The switch is renamed.

To cut switches

- Select the switch(es) you want to cut.
- Select **Cut** from the shortcut menu.

or

- Select **Edit** → **Cut Switches**.

  The switch(es) is/are cut and stored in the clipboard.

To copy switches

- Select the switch(es) you want to copy.
- Select **Copy** from the shortcut menu.

or

- Select **Edit** → **Copy Switches**.

  The selected switch(es) is/are copied to the clipboard.

To add switches

- Select **Paste** from the shortcut menu.

or

- Select **Edit** → **Paste Switches**.

  The switches in the clipboard are added.

  If switches of the same name already exist, you are asked whether you want to continue. For more details see "Adding from the Clipboard" on page 183.

To delete switches

- Select the switch(es) you want to delete.
- Select **Delete** from the shortcut menu.

or

- Select **Edit** → **Delete Switches**.

  The selected switch(es) is/are deleted.

To adjust column width

You can adjust the width of individual columns so that the longest of all character strings in this column can be displayed in entirety. Proceed as follows:

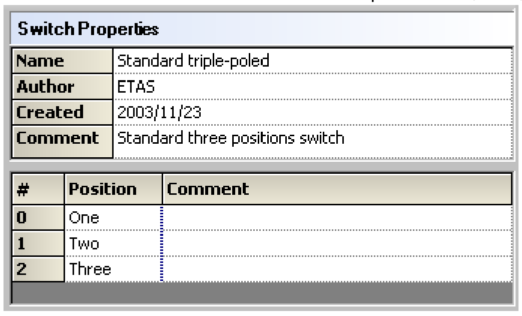
- Position the mouse pointer over the column you want to adjust.
- Select **Adjust Column Width (***Name of column***)** in the shortcut menu.

  The column width is adjusted.

5.3.2 The "Switch Properties" Window

If you select a switch in the "Switches" window just described, all properties of the switch are displayed here.

These include meta data such as "Name", "Author", "Created" and "Comment" and (in the lower part of the window) the description of the switch position like the index of the switch position ("#"), its name and a comment.


Switch Properties

Name	Standard triple-poled
Author	ETAS
Created	2003/11/23
Comment	Standard three positions switch

#	Position	Comment
0	One	
1	Two	
2	Three	

The meta data can be edited directly (by clicking the relevant cell).

Changes such as adding or deleting switch positions are made via a shortcut menu which is described below.

To create a new switch position

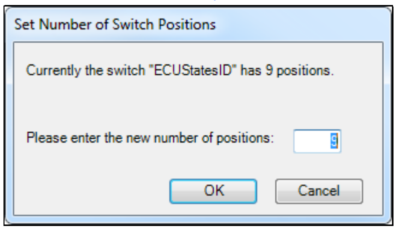
- In the bottom table, select the row under which you want to add the new switch position.
- Select **New Position** from the shortcut menu.

  A new switch position called "Pos_*n*" is created.

To generate several switch positions

- Select **Set Number of Positions** from the shortcut menu.

  The "Set Number of Switch Positions" dialog box opens – this provides information about the current number of switch positions.



Set Number of Switch Positions

Currently the switch "ECUStatesID" has 9 positions.

Please enter the new number of positions: 9

- Enter the total number required.
- Click **OK**.

  The new switch positions are generated.

To duplicate an existing switch position

- Select the row you want to duplicate.
- Select **Duplicate Position** from the shortcut menu.

  The switch position is duplicated. The new name is "*Old name* (1)".

To cut switch position(s)

- Select the switch position(s) you want to cut.
- Select **Cut** from the shortcut menu.

or

- Select **Edit** → **Cut Switch Position(s)**.

  The switch position(s) is/are cut and stored in the clipboard.

To copy switch position(s)

- Select the switch position(s) you want to copy.
- Select **Co py** from the shortcut menu.

or

- Select **Edit** → **Copy Switch Position(s)**.
- The selected switch position(s) is/are copied to the clipboard.

To add switch position(s)

- Select **Paste** from the shortcut menu.

or

- Select **Edit** → **Paste Switch Position(s)**.

  The switch position(s) in the clipboard is/are added.

  If there are already switch positions of the same name, the names are extended with a number – the new name is "*Old name* (1)".

Note

Cutting, copying and pasting also works in exchange with Microsoft Excel tables providing the content of the clipboard is in a maximum of one or two columns.

To delete switch position(s)

- Select the switch position(s) you want to delete.
- Select **Delete** from the shortcut menu.

or

- Select **Edit** → **Delete Switch Position(s)**.

  The selected switch position(s) is/are deleted.

5.3.3 The Main Menu of the Switch Definition File Editor

The Switch Definition File Editor main menu contains the following items:

- **File**

  See "The "File" Menu" on page 180.
- **Edit**

  See "The "Edit" Menu" on page 181.
- **View**

  See "The "View" Menu" on page 181.
- **Help**

  See "The "Help" Menu" on page 181.

The "File" Menu

- **File** → **New**

  Creates a new Switch Definition File.
- **File** → **Open**

  Opens an existing Switch Definition File.
- **File** → **Save**

  Saves the current file.

- **File → Save As**

  Saves the current file under a different file name.
- **File → Recent Files >>**

  Shows the list of the four files last opened from which one can be selected to be opened.
- **File → Exit**

  Exits the Switch Definition File Editor.

The "Edit" Menu

- **Edit → Undo**

  Reverses the last step.
- **Edit → Redo**

  Redoes a step previously undone.
- **Edit → Cut Switch(es)/Switch Positions(s)** *

  Cuts one or more selected switch(es) and stores these in the clipboard.
- **Edit → Copy Cut Switch(es)/Switch Positions(s)** *

  Copies one or more selected switch(es) to the clipboard.
- **Edit → Paste Cut Switch(es)/Switch Positions(s)** *

  Pastes the switch(es) available in the clipboard to the switch list. If a switch with the same name already exists, you can choose between overwriting the existing switch or adding the switch under a different name ("*Old Name(s)*") (see "Adding from the Clipboard" on page 183).
- **Edit → Delete Switch(es)/Switch Positions(s)** *

  Deletes the selected switch(es).

* The menu items depend on whether the "Switches" window or the "Switch Properties" window is active.

The "View" Menu

- **View → Switch Properties**

  Opens/closes the "Switch Properties" window.
- **View → Information Window**

  Opens/closes the "Information" window.

The "Help" Menu

- **Help → User's Guide**

  Opens the English User's Guide (PDF).
- **Help → System Info**

  Opens an information window containing details of the current versions of the components of LABCAR-AUTOMATION 4.2.3.

- **Help → About**

  Opens a window containing general information.

5.3.4 The Toolbar

The toolbar contains the following icons for actions carried out frequently:

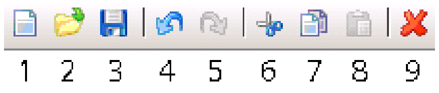


Fig. 5-12 The Toolbar of the Switch Definition File Editor

1. **New**

   Creates a new (empty) Switch Definition File. Exactly the same as **File → New**.
2. **Open**

   Opens a Switch Definition File. Exactly the same as **File → Open**.
3. **Save**

   Saves the Switch Definition File currently open. Exactly the same as **File → Save**.
4. **Undo**

   Reverses the last step. Exactly the same as **Edit → Undo**.
5. **Redo**

   Redoes a step previously undone. Exactly the same as **Edit → Redo**.
6. **Cut Switch(es)**

   Cuts one or more selected switch(es) and stores these in the clipboard. Exactly the same as **Edit → Cut Switch(es)**.
7. **Copy Switch(es)**

   Copies one or more selected switch(es) to the clipboard. Exactly the same as **Edit → Copy Switch(es)**.
8. **Paste Switch(es)**

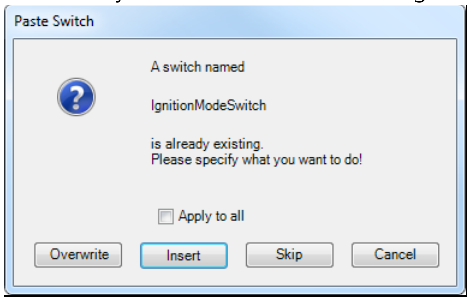
   Adds the switch(es) in the clipboard to the switch list. Exactly the same as **Edit → Paste Switch(es)** (see "Adding from the Clipboard" on page 183).
9. **Delete Switch(es)**

   Deletes the selected switch(es). Exactly the same as **Edit → Delete**.

5.3.5 Adding from the Clipboard

Adding switches from the clipboard makes various actions possible if there is a switch of the same name.

This is why the "Paste Switch" dialog box is opened in such as case.



A switch named

IgnitionModeSwitch

is already existing.
Please specify what you want to do!

This dialog box contains buttons offering the following possibilities:

- **Overwrite**

  The existing switch is overwritten by the switch of the same name in the clipboard.

- **Insert**

  The switch from the clipboard is inserted under a different name ("*Old name* (*n*)").

- **Skip**

  Inserting the current switch from the clipboard is skipped. If there are several switches in the clipboard, the procedure is continued with the next switch.

- **Cancel**

  Inserting is canceled.

Note

If there are several switches in the clipboard, all inserts or overwrites permitted until Cancel are ignored and the complete content of the clipboard is retained.

If there are several switches in the clipboard, selecting the option "Apply to all" results in the procedure selected for the current switch being applied to all other switches (if the names are identical).

5.3.6 The Switch Definition File

The Switch Definition File is an XML file (with the file extension "lcasdt") which contains the definition of all switches to be used. In addition to the known, globally defined Switch Definition File which must be set in the "Templates" window (to be invoked in Test Manager via **Tools** → **Options**), LABCAR-AUTOMATION 4.2.3 now also has "local", project-specific Switch Definition Files. These are intended to make working with individual parameterization requirements easier for the Test Project Manager and the Test Parameter Manager.

The project-specific Switch Definition File is defined in Test Manager via **Project → Options** under "File Locations". The switches defined in the project are displayed in the "Globals" window (see "The "Globals" Window" on page 45) under "Switches".

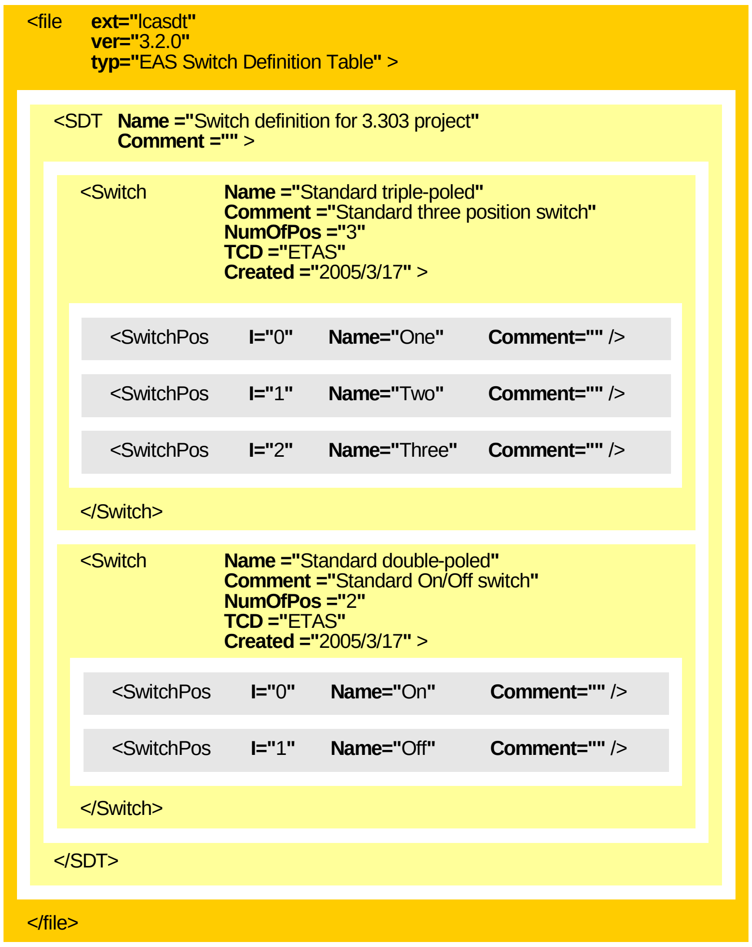
The source of the switches (global or local) is indicated by an icon.

Note

Project-specific ("local") switch definitions overwrite global ones!

5.3.7 The Structure of the Switch Definition File

The example in the following figure shows the typical structure of a Switch Definition File which consists of a freely definable number of "switch" elements which each contain a number of (and at least one) "SwitchPos" elements.



```
<file   ext="lcasdt"
        ver="3.2.0"
        typ="EAS Switch Definition Table" >

  <SDT  Name ="Switch definition for 3.303 project"
        Comment ="" >

    <Switch         Name ="Standard triple-poled"
                    Comment ="Standard three position switch"
                    NumOfPos ="3"
                    TCD ="ETAS"
                    Created ="2005/3/17" >

      <SwitchPos    I="0"    Name="One"      Comment="" />
      <SwitchPos    I="1"    Name="Two"      Comment="" />
      <SwitchPos    I="2"    Name="Three"    Comment="" />

    </Switch>

    <Switch         Name ="Standard double-poled"
                    Comment ="Standard On/Off switch"
                    NumOfPos ="2"
                    TCD ="ETAS"
                    Created ="2005/3/17" >

      <SwitchPos    I="0"    Name="On"       Comment="" />
      <SwitchPos    I="1"    Name="Off"      Comment="" />

    </Switch>

  </SDT>

</file>
```

Fig. 5-13 The Structure of a Switch Definition File

The following table contains attributes, their type and meaning of the different XML elements.

Attribute	Type	Meaning
Name	String	Currently not used
Comment	String	Currently not used

Tab. 5-1 Attributes of the SDT Element

Attribute	Type	Meaning
Name	String	Name of the switch
Comment	String	A comment
NumOfPos	Integer	Number of switch positions (≥ 1)
TCD	String	Author of the switch
Created	Date	Date created

Tab. 5-2 Attributes of the Switch Element

Attribute	Type	Meaning
I	Integer	Index of the switch position*
Name	String	Name of the switch position
Comment	String	A comment on the switch position
* The numbers of the switch position must start at 0 and must not have any gaps.		

Tab. 5-3 Attributes of the SwitchPos Element

5.4 The UuT List Editor

The UuT List Editor is used to view and edit UuT Lists (file extension "dbu").

5.4.1 The UuT List

A UuT List is a collection of UuTs which are available in the project. A UuT is characterized by its name, the "UuT Description" and the "UuT Environment".

In the UuT List Editor, you manage this collection of UuTs by adding and deleting UuTs and by changing properties such as Description or Environment.

In the UuT List Editor you can create, rename and delete Descriptions and Environments. You can also assign these to one or more UuTs in your list.

UuT Description

With a UuT of the type "ECU" (and this is currently the only type supported), this field contains a "Device" which in turn can contain any number of processors. In addition, further program and other data can be specified for this ECU.

UuT Environment

This field mainly displays a range of what are referred to as "ID strings". These are of significance for the configuration of the test bench for the selected UuT.

The values of these strings can be edited in the UuT List Editor – e.g. when the settings provided with the "Global UuT List" do not correspond to the current tool configuration.

To open the UuT List Editor

- Select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2** → **Editors** → **UuT List Editor**.

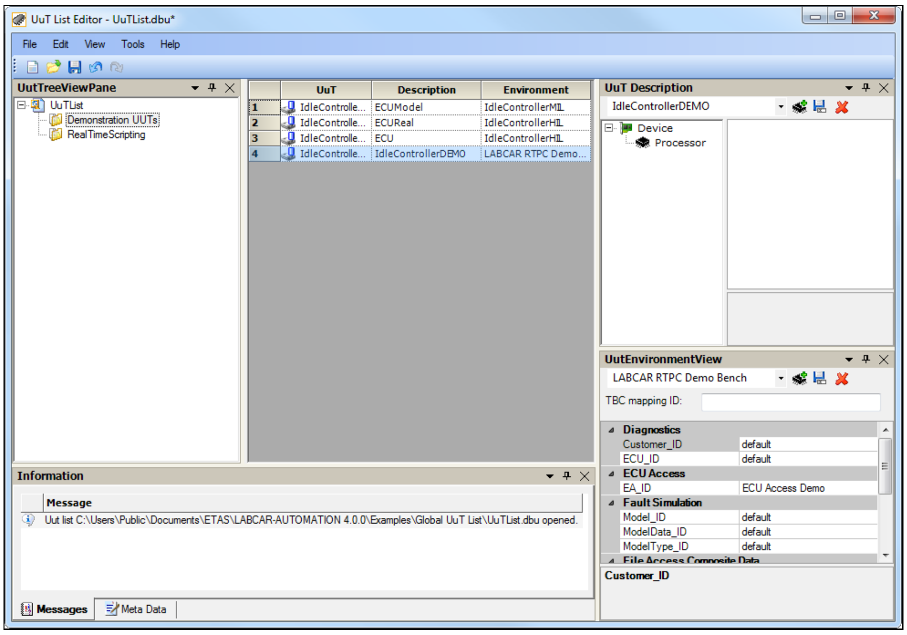
  The UuT List Editor opens and an new UuT List is generated.

You can now add UuTs to this list – but you can also open existing UuT Lists and edit them accordingly.

To open a UuT List

- Select **File** → **Open** to open an existing UuT List.

Fig. 5-14 shows the window of the UuT List Editor with an open file.



	UuT	Description	Environment
1	IdleControlle...	ECUModel	IdleControllerMIL
2	IdleControlle...	ECUReal	IdleControllerHIL
3	IdleControlle...	ECU	IdleControllerHIL
4	IdleControlle...	IdleControllerDEMO	LABCAR RTPC Demo...

Diagnostics	
Customer_ID	default
ECU_ID	default
ECU Access	
EA_ID	ECU Access Demo
Fault Simulation	
Model_ID	default
ModelData_ID	default
ModelType_ID	default
File Access Composite Data	

Customer_ID

```
Message
Uut list C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Global UuT List\UuTList.dbu opened.
```

Fig. 5-14 The User Interface of the UuT List Editor

The user interface of the UuT List Editor consists of the following parts:

- UuT Tree

  This window shows the hierarchy of the UuTs in the UuT List.

- List of UuTs

  All UuTs defined in this file are listed here with the names of the "UuT Description" and "UuT Environment".

  For a description of the list of UuTs refer to the section "The List of the UuTs" on page 190.

- Description

  All data on the UuT Description can be edited here. For more details, refer to the section "Processing and Managing UuT Descriptions/Environments" on page 192.

- Environment

  All data on the UuT Description can be edited here. For more details, refer to the section "Processing and Managing UuT Descriptions/Environments" on page 192.

- The "Information" Window

  This is the window familiar from all LABCAR-AUTOMATION applications in which information and error messages are issued.

5.4.2 The Main Menu of the UuT List Editor

This section contains a short description of the main menu of the UuT List Editor.

"File" Menu

- **File → New**

  Creates a new (empty) UuT List.
- **File → Open**

  Opens an existing UuT List.
- **File → Close**

  Closes the UuT List currently open.
- **File → Save**

  Saves the UuT List currently open.
- **File → Save as**

  Saves the UuT List currently open under a different name.
- **File → Recent Files →**

  Shows the list of the four files last opened from which one can be selected to be opened.
- **File → Exit**

  Exits the UuT List Editor.

"Edit" Menu

- **Edit → Undo**

  Reverses your last step.
- **Edit → Redo**

  Repeats a step you have just reversed.

"View" Menu

- **View → Information Window**

  Shows (or hides) the Information Window.
- **View → UuT Tree**

  Shows (or hides) the "UuT Tree" window.
- **View → UuT Description**

  Shows (or hides) the "UuT Description" window.
- **View → UuT Environment**

  Shows (or hides) the "UuT Environment" window.

"Tools" Menu

- **Tools → Options**

  Opens the "Options" window (see "Saving Options" on page 195).

"Help" Menu

- **Help → User's Guide**

  Opens the English User's Guide (PDF).
- **Help → System Info**

  Opens an information window containing details of the current versions of the components of LABCAR-AUTOMATION 4.2.3.
- **Help → About**

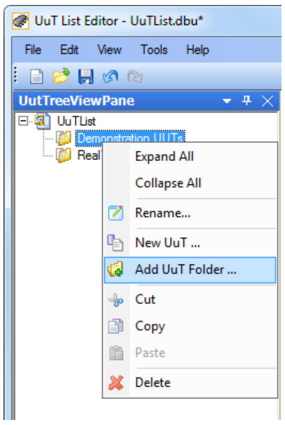
  Opens a window containing general information.

5.4.3 The UuT Tree

In the UuT Tree, the collection of UuTs in the UuT List can be assigned a folder structure.

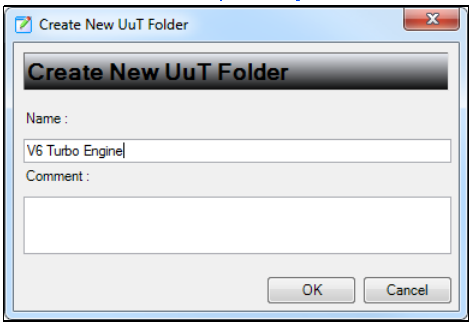
To add a folder

- Right-click the list (or the folder) to which you want to add a folder.
- Select **Add UuT Folder** from the shortcut menu.



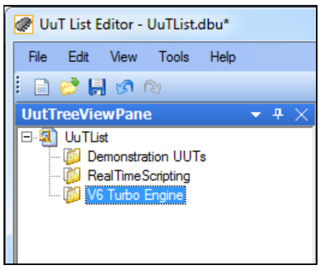
The "Create New UuT Folder" window opens.

- Enter a name (and possibly a comment).



V6 Turbo Engine

The folder is created and shown in the UuT Tree.



UuTList
- Demonstration UUTs
- RealTimeScripting
- V6 Turbo Engine

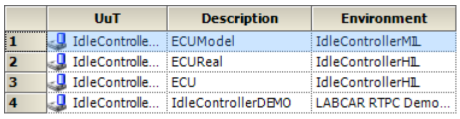
UuTs added in the Test Manager (**Add Uut from list**) are also shown with this hierarchy in the UuT Explorer.

You can also add UuTs in the UuT Tree – the mechanism is the same as adding to the list of UuTs (see "To add a UuT to the list" on page 190).

5.4.4 The List of the UuTs

This list shows all UuTs of the UuT List with name, UuT Description and UuT Environment.

The UuTs are listed with their name, UuT Description and UuT Environment.

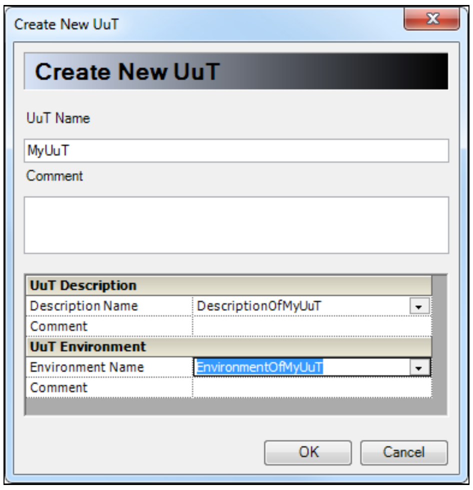


	UuT	Description	Environment
1	IdleControlle...	ECUModel	IdleControllerMIL
2	IdleControlle...	ECUReal	IdleControllerHIL
3	IdleControlle...	ECU	IdleControllerHIL
4	IdleControlle...	IdleControllerDEMO	LABCAR RTPC Demo...

To add a UuT to the list

- Select **New UuT** in the UuT List's shortcut menu.

  The "Create New UuT" window opens.
- Enter the name for the UuT ("UuT Name"), a name for the description ("UuT Description") and a name for the environment ("UuT Environment").



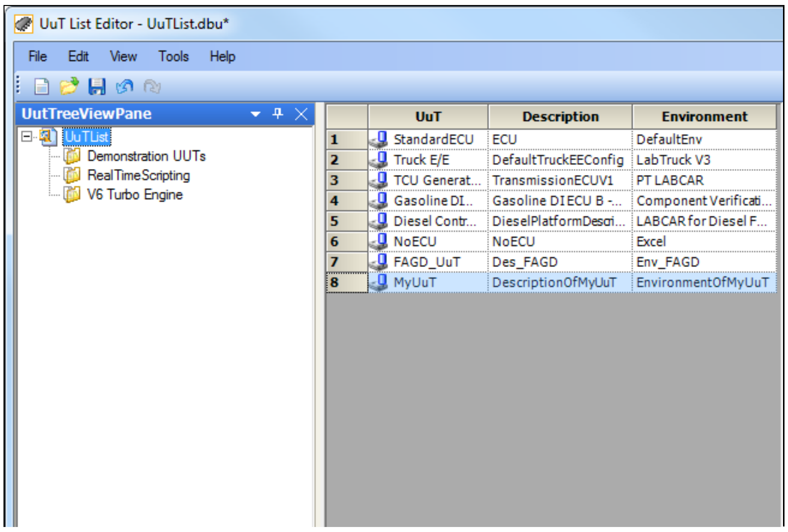
UuT Name: MyUuT

Comment:

UuT Description	
Description Name	DescriptionOfMyUuT
Comment	
UuT Environment	
Environment Name	EnvironmentOfMyUuT
Comment	

- Click **OK**.

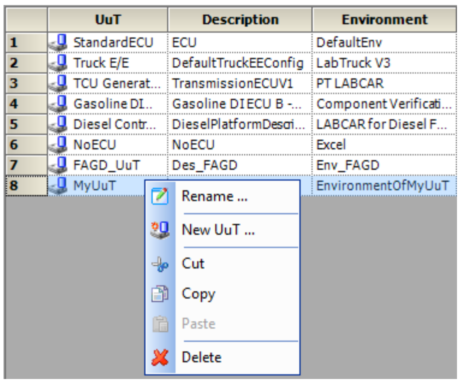
  The UuT is added to the list.



	UuT	Description	Environment
1	StandardECU	ECU	DefaultEnv
2	Truck E/E	DefaultTruckEEConfig	LabTruck V3
3	TCU Generat...	TransmissionECUV1	PT LABCAR
4	Gasoline DI...	Gasoline DI ECU B -...	Component Verificati...
5	Diesel Contr...	DieselPlatformDescri...	LABCAR for Diesel F...
6	NoECU	NoECU	Excel
7	FAGD_UuT	Des_FAGD	Env_FAGD
8	MyUuT	DescriptionOfMyUuT	EnvironmentOfMyUuT

To rename a UuT

- Select the UuT to be renamed from the list of UuTs.
- Select **Rename** from the shortcut menu.



	UuT	Description	Environment
1	StandardECU	ECU	DefaultEnv
2	Truck E/E	DefaultTruckEEConfig	LabTruck V3
3	TCU Generat...	TransmissionECUV1	PT LABCAR
4	Gasoline DI...	Gasoline DI ECU B -...	Component Verificati...
5	Diesel Contr...	DieselPlatformDescri...	LABCAR for Diesel F...
6	NoECU	NoECU	Excel
7	FAGD_UuT	Des_FAGD	Env_FAGD
8	MyUuT		EnvironmentOfMyUuT

  The “Rename UuT” window opens.

- Enter a new name and possibly a new comment.

If, for example, you want to add a UuT which is only very slightly different from an existing one, you can do this by copying and pasting it to the list of UuTs.

You can also cut a UuT from the list and add it again to the UuT Tree at a different structure level.

To copy and paste UuTs

- Right-click the desired UuT in the list.
- Select **Copy** from the shortcut menu.
- To add the same UuT back to the list, select **Paste** from the shortcut menu.

  The UuT is added with the new name "Copy of ...".

- To assign the copied (or cut) UuT to a (different) structure level in the UuT Tree, select the desired level with the mouse.
- Right-click it and select **Paste** from the shortcut menu.

  The UuT is reinserted at the selected structure level.

To delete a UuT from the list

- Select the UuT to be deleted from the list of UuTs.
- Select **Delete** from the shortcut menu.

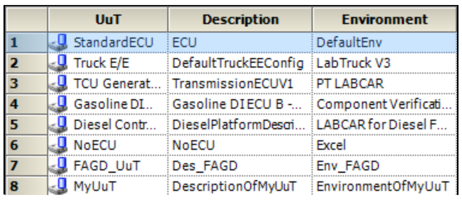
  The selected UuT is deleted.

5.4.5 Processing and Managing UuT Descriptions/Environments

This section contains a description of how to assign UuT Descriptions and UuT Environments to a UuT, create, rename and delete them.

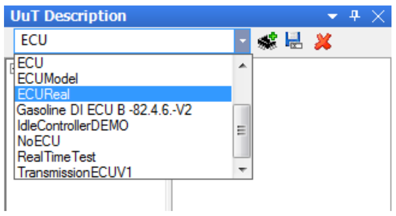
To assign an existing UuT Description/UuT Environment to the UuT

- From the list, select the UuT to which you want to assign a Description.



	UuT	Description	Environment
1	StandardECU	ECU	DefaultEnv
2	Truck E/E	DefaultTruckEEConfig	LabTruck V3
3	TCU Generat...	TransmissionECUV1	PT LABCAR
4	Gasoline DI..	Gasoline DIECU B -...	Component Verificati...
5	Diesel Contr...	DieselPlatformDescri...	LABCAR for Diesel F...
6	NoECU	NoECU	Excel
7	FAGD_UuT	Des_FAGD	Env_FAGD
8	MyUuT	DescriptionOfMyUuT	EnvironmentOfMyUuT

- Select one of the available Descriptions in the "UuT Description" window.



  The selected Description is assigned to the UuT.

- Proceed in exactly the same way to assign a UuT Environment.

You can also assign a new UuT Description or a new UuT Environment to a selected UuT by duplicating existing ones and changing them to suit your specifications.

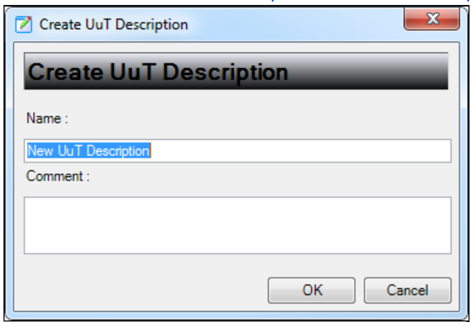
To create a new UuT Description/new UuT Environment

- Select the UuT for which you want to create a new Description.
- Assign the UuT the Description which is most similar to the one to be created.

  This means the "newly" created Description is a copy of the one assigned to date.
- If necessary, remove the assigned UuT Description (see "To remove/delete a UuT Description/UuT Environment" on page 194).

  This results in the created Description being "empty".
- Click the **Create New Description** icon.

  The "Create UuT Description" window opens.



Create UuT Description

Create UuT Description

Name :

New UuT Description

Comment :

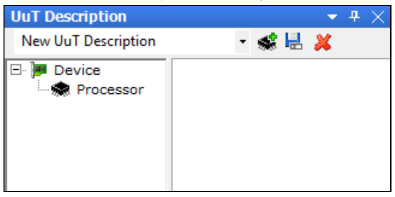
- Enter a name (and possibly a comment) for the new UuT Description.

Note

The names must be unique!
Already existing names are shown in red with a warning sign and cannot be assigned.

- Click **OK**.

  The UuT Description is created and assigned to the UuT selected previously.

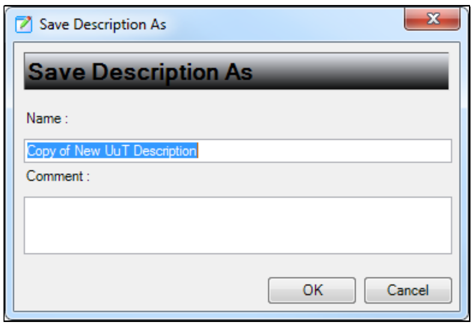


- Proceed in exactly the same way to create a UuT Environment.

To rename a UuT Description/UuT Environment

- Select the UuT whose Description is to be renamed.
- Click the **Save Description As** icon.

  The "Save Description As" window opens.



Save Description As

Save Description As

Name :

Copy of New UuT Description

Comment :

- Enter a new name (and possibly a comment).

  The UuT Description is renamed - the assignment to the selected UuT remains (although under a new name).
- Proceed in exactly the same way to rename a UuT Environment.

To remove/delete a UuT Description/UuT Environment

- Select the UuT whose Description is to be removed/ deleted.

Note

The Description is only deleted if it is exclusively assigned to the selected UuT. If there are several UuTs to which this Description has been assigned, the assignment to this UuT is canceled.

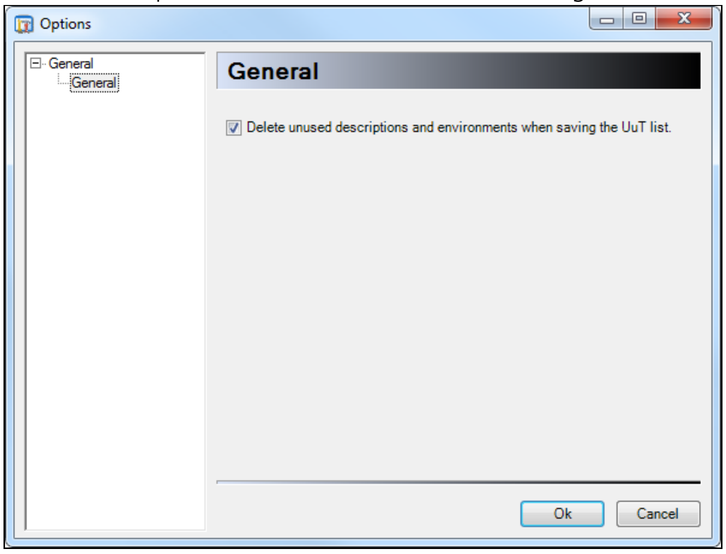
- Click the **Delete Description** icon.

  The Description is deleted from the list of available Descriptions. If it (see above) was assigned otherwise, a warning will be displayed in the "Messages" window, that the description could not be deleted.
- Proceed in exactly the same way to remove/delete a UuT Environment.

5.4.6 Saving Options

A UuT List is saved as an XML file with the file extension "dbu". Parallel to this file, two directories, `Descriptions` and `Environment`, are created which contain the UuT Descriptions and UuT Environments respectively of the UuTs of the list.

If you want UuT Descriptions and UuT Environments which are not used in the project to be deleted, select Tools → Options and activate the option "Delete unused descriptions and environments when saving the UuT List".



☑ Delete unused descriptions and environments when saving the UuT list.

6 Tutorial

This tutorial uses a simple experiment to introduce you to working with LABCAR-AUTOMATION 4.2.3.

For details of the "IdleController" model, refer to the Tutorial in "LABCAR-OPERATOR V5.3.1 - Getting Started", which is part of the delivery scope of LABCAR-OPERATOR as a PDF (on the installation CD).

Structure of this Tutorial

This tutorial is structured as follows:

- "Before you Begin" on page 198

  Before you start this tutorial, please read the notes in this section.
- "Running a Test Project" on page 199

  At the start of this tutorial, you are going to run a project as test handler and evaluate the results of the test report. The creation of this project is demonstrated in section "Creating a Test Project" on page 211.
- "Creating a Test Project" on page 211

  The aim of this section is to recreate the LABCAR-AUTOMATION project which has just been executed in the Test Manager.
- "Configuring the Test Bench" on page 233

  This section contains a short introduction on how to view the configuration files used in the tutorial project.

6.1 Before you Begin

Before you begin, please read the following tips and comments.

6.1.1 Installing LABCAR-AUTOMATION 4.2.3

How to install LABCAR-AUTOMATION 4.2.3 and the system requirements are described in detail in the chapter "Installation" on page 27.

6.1.2 Installing LABCAR-OPERATOR and LABCAR-RTPC

To work with this tutorial, you need LABCAR-OPERATOR 5.2.1 and the corresponding version of LABCAR-RTPC.

6.1.3 Installing Microsoft Visual Studio

To create test cases in C# or C++, you can use one of the following versions:

- Visual Studio 2012/2013 Professional Edition

  The integrated development environment by Microsoft

or

- Visual Studio 2012/2013 Express Edition

  The free starter version

6.1.4 The Data Directory of LABCAR-AUTOMATION

When working with LABCAR-AUTOMATION projects, it is useful to be familiar with the significance and function of the individual files that belong to this kind of project. For a general description of the data directory, please refer to section 2.4.1 on page 37.

6.2 Running a Test Project

At the start of this tutorial, you are going to run a project as test handler and evaluate the results of the test report. The creation of this project is demonstrated in section "Creating a Test Project" on page 211.

You will carry out the following steps:

- "To launch Test Handler" on page 199
- "To open a project" on page 199
- "To synchronize test cases in the Test Manager" on page 200
- "To provide a test bench configuration" on page 201
- "To specify a directory for test reports" on page 202
- "To run a test project" on page 203
- "To open a test report" on page 204
- "To filter a test report" on page 205
- "To evaluate a test report" on page 206

To launch Test Handler

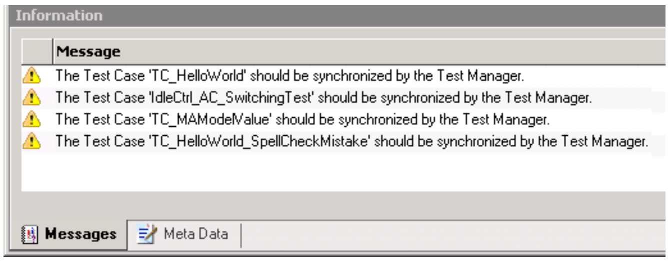
- In the Windows Start menu, select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2.3** → **Test Handler**.

  Test Handler is launched.

To open a project

- Select **File** → **Open Project**.
- In the file selector window, select the file `<`*Drive*`>:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2.3\Examples\LCA Test Projects\Test Project IdleController\Test Project IdleController.lcaprj`.

  The project opens in Test Handler.
- If warnings are issued in the “Information” window (“Messages” tab) (see figure), proceed as follows.



Message
The Test Case 'TC_HelloWorld' should be synchronized by the Test Manager.
The Test Case 'IdleCtrl_AC_SwitchingTest' should be synchronized by the Test Manager.
The Test Case 'TC_MAModelValue' should be synchronized by the Test Manager.
The Test Case 'TC_HelloWorld_SpellCheckMistake' should be synchronized by the Test Manager.

To synchronize test cases in the Test Manager

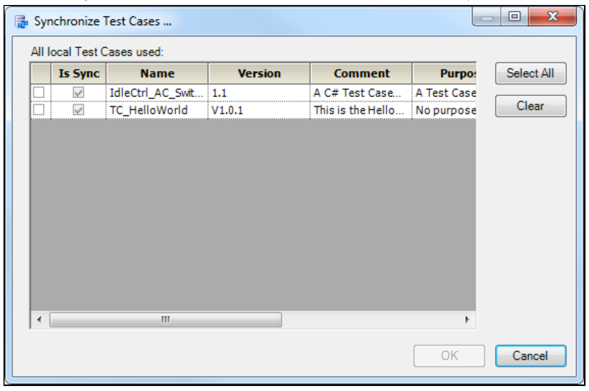
- In the Windows Start menu, select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2.3** → **Test Manager**.

  Test Manager is launched.
- Select **File** → **Open** and open the top project.
- In the Project Explorer ("Project View" tab, select the UuT Group "Release Tests for Components".
- Select **Project** → **Synchronize** → **Test Cases**

or

- Click the **Test Cases** icon.

  The "Synchronize Test Cases" window opens.



Synchronize Test Cases ...

All local Test Cases used:

	Is Sync	Name	Version	Comment	Purpo:
☐	✓	IdleCtrl_AC_Swit...	1.1	A C# Test Case...	A Test Case
☐	✓	TC_HelloWorld	V1.0.1	This is the Hello...	No purpose

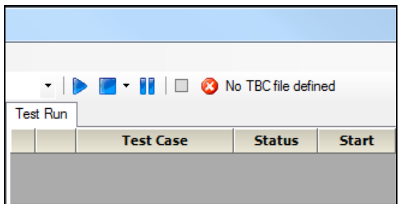
  This window shows all test cases which are assigned to the UuT Group.
- Select all test cases using **Select All**.
- Click **OK**.

  The test cases are synchronized with the current Test Release Library (see "Information" window).
- Save the project with **Save**.
- To close the Test Manager select **File** → **Exit**.
- Return to the Test Handler.

Note

The Test Handler detects that changes have been made to the project currently open and prompts you to agree to a project refresh.

"No TBC file defined" in the toolbar shows that the project has not yet been assigned a Test Bench Configuration.



If a Test Bench Configuration has already been assigned, still follow the instructions below to ensure that the correct file has been assigned.

The test bench configuration (*.tbc) is the list of tools available on this test system – to find out more on how to edit this kind of file, please refer to the section "The Test Bench Configuration File" on page 233.

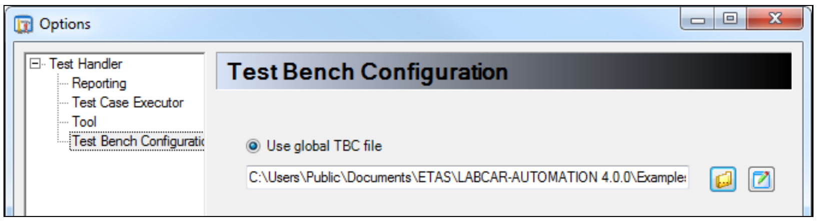
To provide a test bench configuration

- Click the "No TBC file defined" icon.

or

- In the main menu, select **Tools** → **Options**.

  The window for setting all options opens.
- Select "Test Bench Configuration" and activate the option "Use global TBC file".

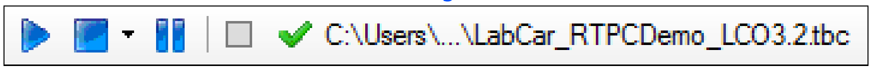


Use global TBC file

C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Example:

- Click the folder icon.
- Select <*Drive*>`:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2.3\Examples\Test Bench Configurations\Demo Test Bench\LCO 5.2\LabCar_RTPCDemo.tbc`.
- Click **OK**.

  The toolbar shows that a test bench configuration has been assigned.



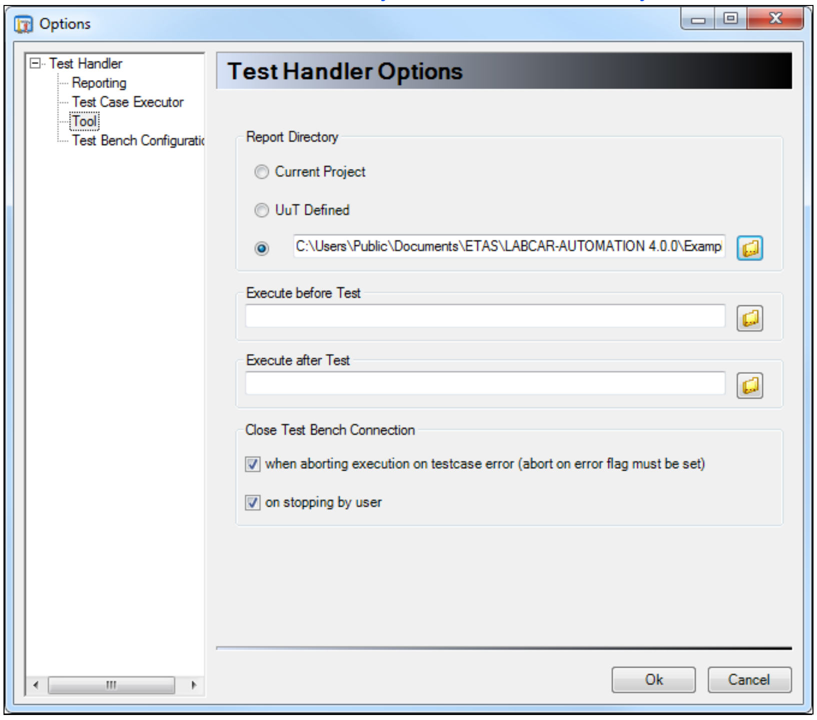
C:\Users\...\LabCar_RTPCDemo_LCO3.2.tbc

To save and manage all test reports somewhere central, specify a corresponding directory.

To specify a directory for test reports

- In the main menu, select **Tools** → **Options**.

  The window for setting all options opens.
- Select “Tool”.
- In the “Report Directory” field activate the option that enables you to select a directory.



Report Directory
- ( ) Current Project
- ( ) UuT Defined
- (●) C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examp

Execute before Test

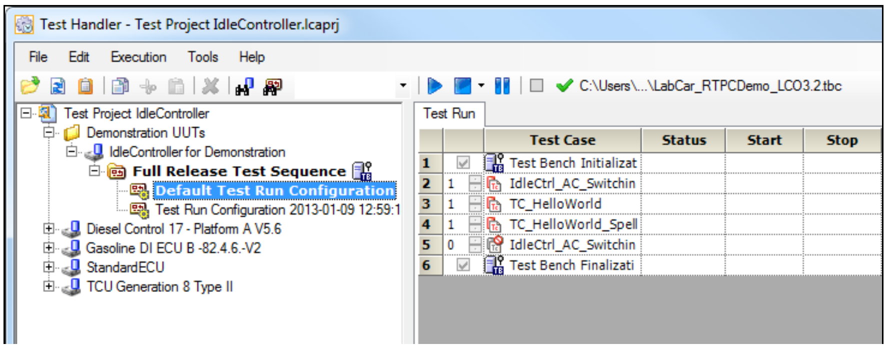
Execute after Test

Close Test Bench Connection
- [✓] when aborting execution on testcase error (abort on error flag must be set)
- [✓] on stopping by user

- Click the folder icon.
- In the file selector window, select the folder <*Drive*>:`\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2.3\Examples\Test Reports`.
- Close the "Test Handler Options" window with **OK**.

To run a test project

- In the project tree, open the “Demonstration UUTs” folder, select the “IdleController for Demonstration” UuT, the “Full Release Test Sequence” in it and then the “Default Test Run Configuration” from that.


Test Handler - Test Project IdleController.lcaprj
File
Edit
Execution
Tools
Help
C:\Users\...\LabCar_RTPCDemo_LCO3.2.tbc
Test Run

Test Project IdleController
- Demonstration UUTs
  - IdleController for Demonstration
    - **Full Release Test Sequence**
      - **Default Test Run Configuration**
      - Test Run Configuration 2013-01-09 12:59:1
- Diesel Control 17 - Platform A V5.6
- Gasoline DI ECU B -82.4.6.-V2
- StandardECU
- TCU Generation 8 Type II

		Test Case	Status	Start	Stop
1	✓	Test Bench Initializat			
2	1	IdleCtrl_AC_Switchin			
3	1	TC_HelloWorld			
4	1	TC_HelloWorld_Spell			
5	0	IdleCtrl_AC_Switchin			
6	✓	Test Bench Finalizati			

The table on the right-hand side displays the test functionality, and test bench initialization or finalization respectively assigned to this sequence.

- To start the test, click the **Start Test Run** icon.

or

- Select **Execution** → **Start Test Run**.

or

- Press <F5>.

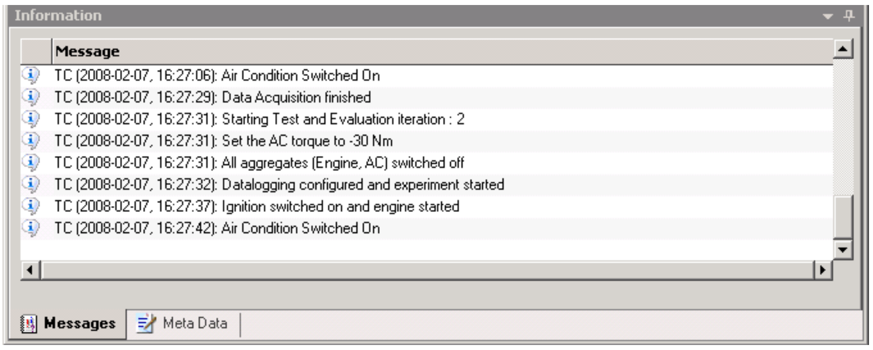
  The test starts:

  – LABCAR-EE is launched

  – The project is loaded

  – The corresponding model opens

  – The latest messages are displayed in the “Information” window.

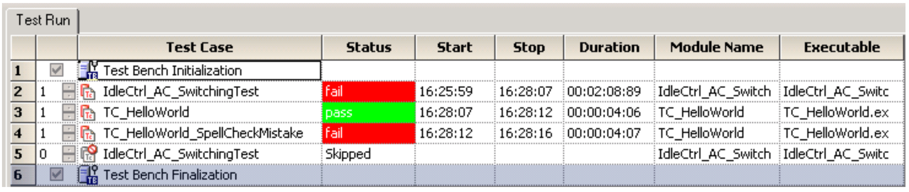


Message
TC (2008-02-07, 16:27:06): Air Condition Switched On
TC (2008-02-07, 16:27:29): Data Acquisition finished
TC (2008-02-07, 16:27:31): Starting Test and Evaluation iteration : 2
TC (2008-02-07, 16:27:31): Set the AC torque to -30 Nm
TC (2008-02-07, 16:27:31): All aggregates (Engine, AC) switched off
TC (2008-02-07, 16:27:32): Datalogging configured and experiment started
TC (2008-02-07, 16:27:37): Ignition switched on and engine started
TC (2008-02-07, 16:27:42): Air Condition Switched On

  – During execution, a test report is created and constantly updated.

Once the test is completed, the LABCAR-EE operating window is closed again.

The results of the individual test cases are then shown in the Test Handler “Status” column.

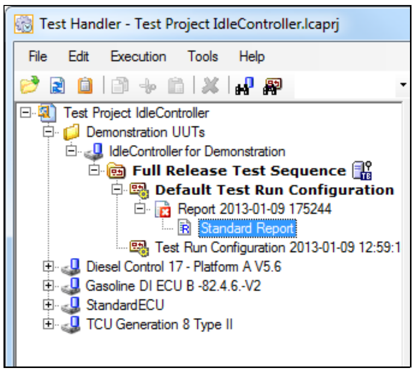

Test Run

		Test Case	Status	Start	Stop	Duration	Module Name	Executable
1	✓	Test Bench Initialization						
2	1	IdleCtrl_AC_SwitchingTest	fail	16:25:59	16:28:07	00:02:08:89	IdleCtrl_AC_Switch	IdleCtrl_AC_Switc
3	1	TC_HelloWorld	pass	16:28:07	16:28:12	00:00:04:06	TC_HelloWorld	TC_HelloWorld.ex
4	1	TC_HelloWorld_SpellCheckMistake	fail	16:28:12	16:28:16	00:00:04:07	TC_HelloWorld	TC_HelloWorld.ex
5	0	IdleCtrl_AC_SwitchingTest	Skipped				IdleCtrl_AC_Switch	IdleCtrl_AC_Switc
6	✓	Test Bench Finalization						

The test is evaluated in the Report Viewer.

To open a test report

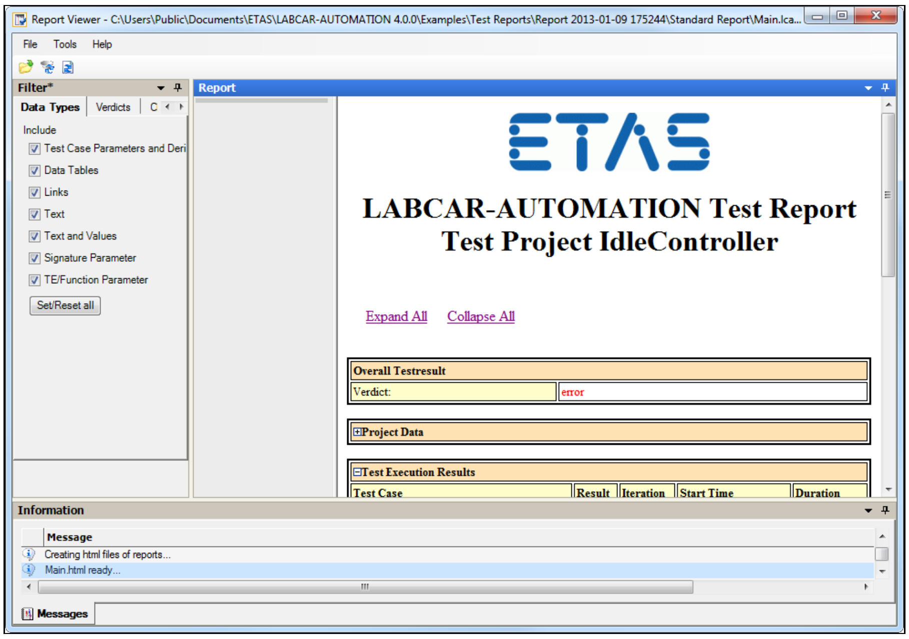
- Double-click “Standard Report” under the report in the Project Explorer.


Test Handler - Test Project IdleController.lcaprj

```
Test Project IdleController
  Demonstration UUTs
    IdleController for Demonstration
      Full Release Test Sequence
        Default Test Run Configuration
          Report 2013-01-09 175244
            Standard Report
        Test Run Configuration 2013-01-09 12:59:1
  Diesel Control 17 - Platform A V5.6
  Gasoline DI ECU B -82.4.6.-V2
  StandardECU
  TCU Generation 8 Type II
```

Report Viewer is launched and the report opened.

- Click the hyperlink “Overview” to display the overview page.


Report Viewer - C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Reports\Report 2013-01-09 175244\Standard Report\Main.lca...

Include
- ✓ Test Case Parameters and Deri
- ✓ Data Tables
- ✓ Links
- ✓ Text
- ✓ Text and Values
- ✓ Signature Parameter
- ✓ TE/Function Parameter

ETAS

LABCAR-AUTOMATION Test Report
Test Project IdleController

Overall Testresult	
Verdict:	error

⊞Project Data

⊟Test Execution Results

Test Case	Result	Iteration	Start Time	Duration

Information

Message
Creating html files of reports...
Main.html ready...

- Click the hyperlink "1. IdleCtrl_AC_SwitchingTest_1" to display detailed information on the test run.

For detailed information on test reports, refer to the section "Test Reports" on page 125.

The content of a test report comes from two sources:

- From the tool itself, e.g. signature calls ("implicit logging")
- From the report text defined by the Test Case Developer or the report form ("explicit logging")

To make the display less detailed, but clearer to view, you can load a view configuration which contains specific filter settings. This is optional.

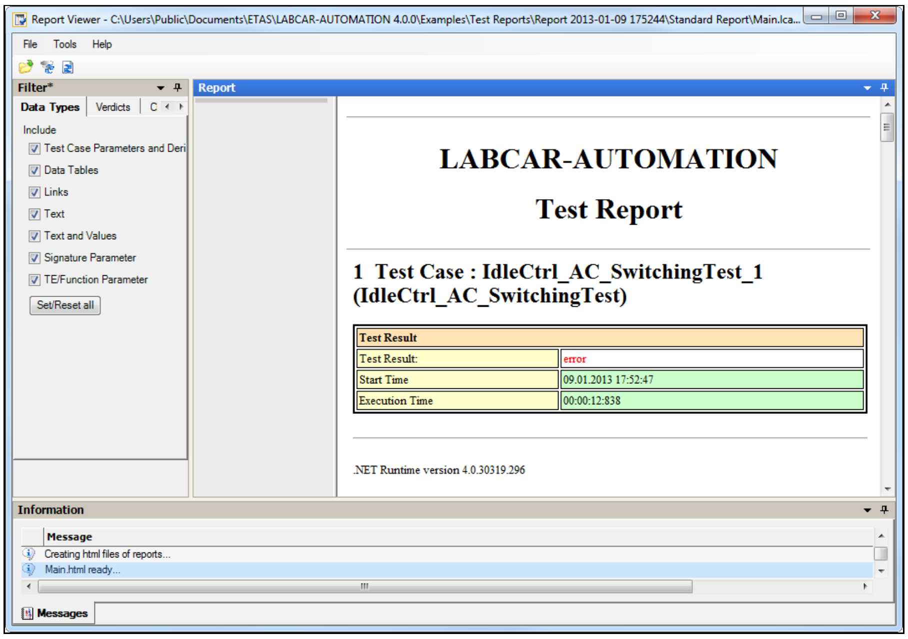
To filter a test report

- Select **File** → **Open View Configuration**.
- Open the directory `<`*Drive*`>:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2.3\Examples\Test Reports\Report View Configurations`.
- Select the `DemoReportOverview.lcarvc` file and click **OK**.

  The view configuration is loaded.

To evaluate a test report

- In the overview, click the first test case of the test run “IdleCtrl_AC_SwitchingTest_1”.



Report Viewer - C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Reports\Report 2013-01-09 175244\Standard Report\Main.lca...

File Tools Help

Filter* — Data Types | Verdicts | C

Include
- [✓] Test Case Parameters and Deri
- [✓] Data Tables
- [✓] Links
- [✓] Text
- [✓] Text and Values
- [✓] Signature Parameter
- [✓] TE/Function Parameter

Set/Reset all

Report

LABCAR-AUTOMATION

Test Report

1 Test Case : IdleCtrl_AC_SwitchingTest_1 (IdleCtrl_AC_SwitchingTest)

Test Result	
Test Result:	error
Start Time	09.01.2013 17:52:47
Execution Time	00:00:12:838

.NET Runtime version 4.0.30319.296

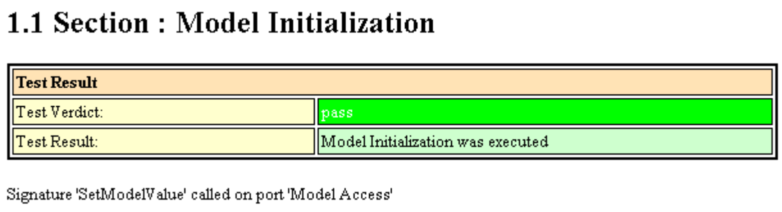
Information

Message
Creating html files of reports...
Main.html ready...

Messages

It contains the “Model Initialization” and – with the idle controller model – the three times a specific test functionality was run within a test case.

“Model Initialization” ends with the verdict “pass”.

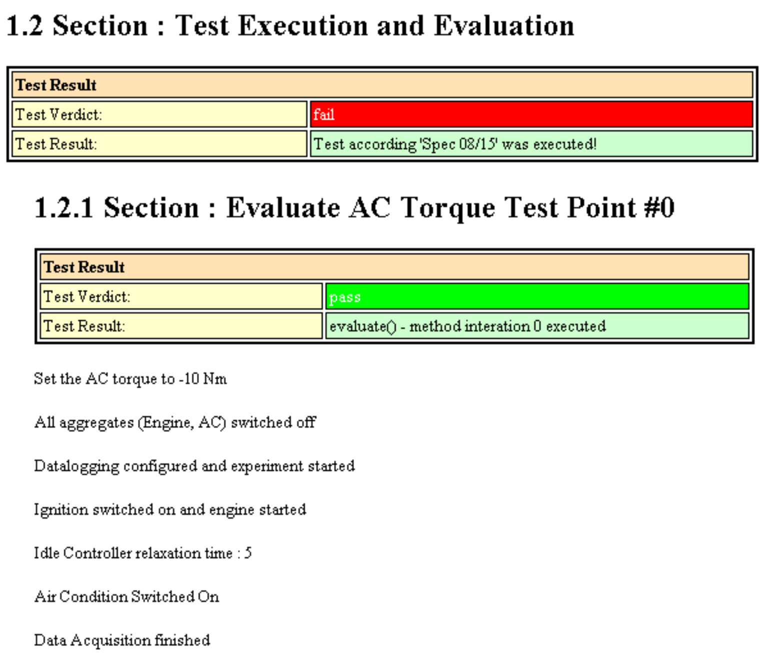


1.1 Section : Model Initialization

Test Result	
Test Verdict:	pass
Test Result:	Model Initialization was executed

Signature 'SetModelValue' called on port 'Model Access'

"1.2 Test Execution and Evaluation" is assigned the verdict "fail" because one or more parts of it are assigned the verdict "fail".



1.2 Section : Test Execution and Evaluation

Test Result	
Test Verdict:	fail
Test Result:	Test according 'Spec 08/15' was executed!

1.2.1 Section : Evaluate AC Torque Test Point #0

Test Result	
Test Verdict:	pass
Test Result:	evaluate() - method interation 0 executed

Set the AC torque to -10 Nm

All aggregates (Engine, AC) switched off

Datalogging configured and experiment started

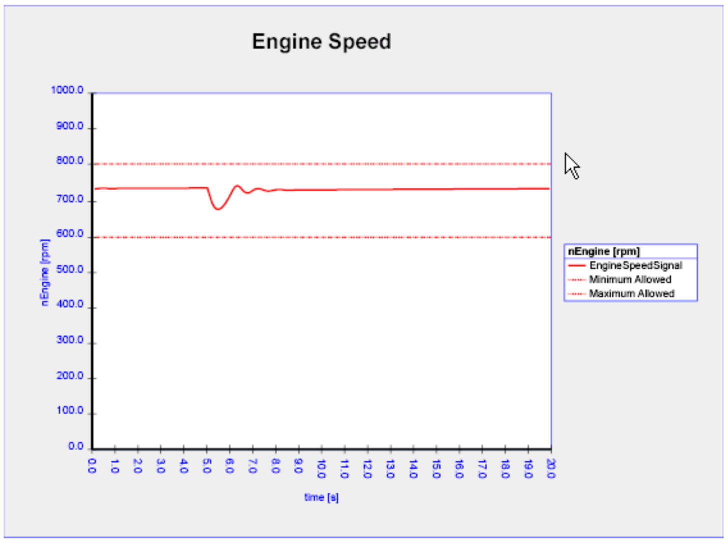
Ignition switched on and engine started

Idle Controller relaxation time : 5

Air Condition Switched On

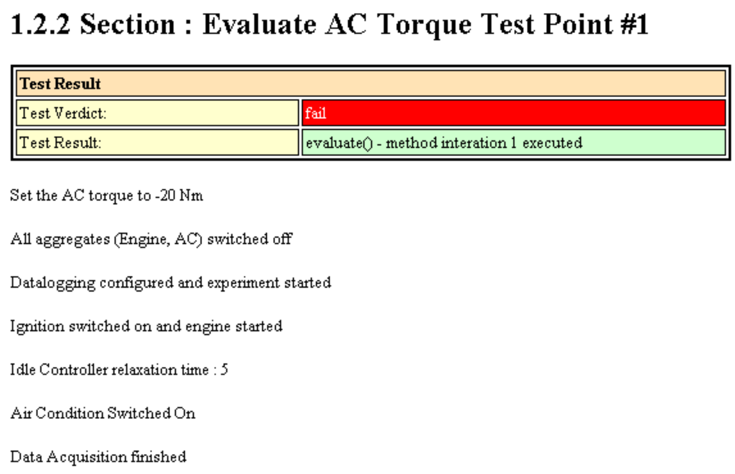
Data Acquisition finished

In the first part ("1.2.1 Section: Evaluate AC Torque Test Point #0"), the parameter for the air conditioning torque ("AirConditionTorque") is set to "-10 Nm". The deviations in speed resulting from the air conditioning being switched on remain within the permissible speed range.



This run is therefore assigned the verdict "pass".

In the second part (“1.2.2 Section: Evaluate AC Torque Test Point #1”), the parameter for the air conditioning torque (“AirConditionTorque”) is set to “-20 Nm”.



1.2.2 Section : Evaluate AC Torque Test Point #1

Test Result	
Test Verdict:	fail
Test Result:	evaluate() - method interation 1 executed

Set the AC torque to -20 Nm

All aggregates (Engine, AC) switched off

Datalogging configured and experiment started

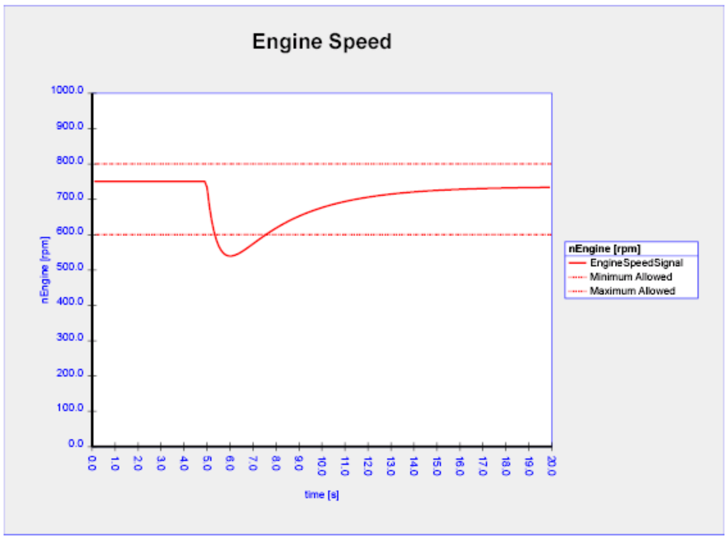
Ignition switched on and engine started

Idle Controller relaxation time : 5

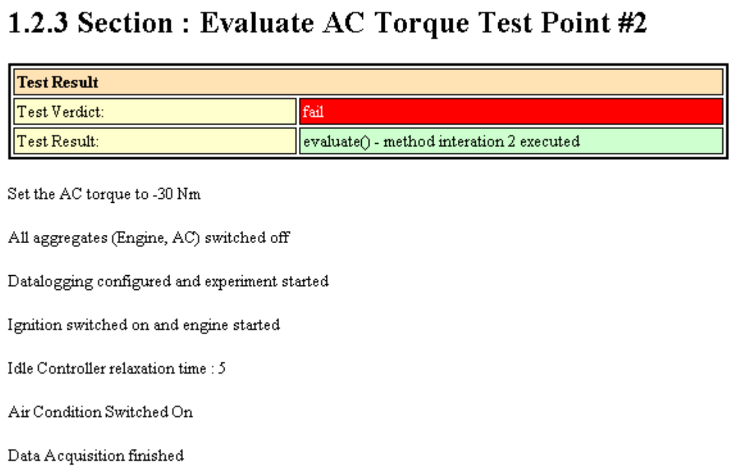
Air Condition Switched On

Data Acquisition finished

The engine speed recovers from the drop caused by the air conditioning being switched on, but the minimum speed is outside the permissible speed range. It is assigned the verdict “fail”.



In the third part (“1.2.3 Section: Evaluate AC Torque Test Point #2”) the parameter for the air conditioning torque (“AirConditionTorque”) is set to “-30 Nm”.



1.2.3 Section : Evaluate AC Torque Test Point #2

Test Result	
Test Verdict:	🟥 fail
Test Result:	evaluate() - method interation 2 executed

Set the AC torque to -30 Nm

All aggregates (Engine, AC) switched off

Datalogging configured and experiment started

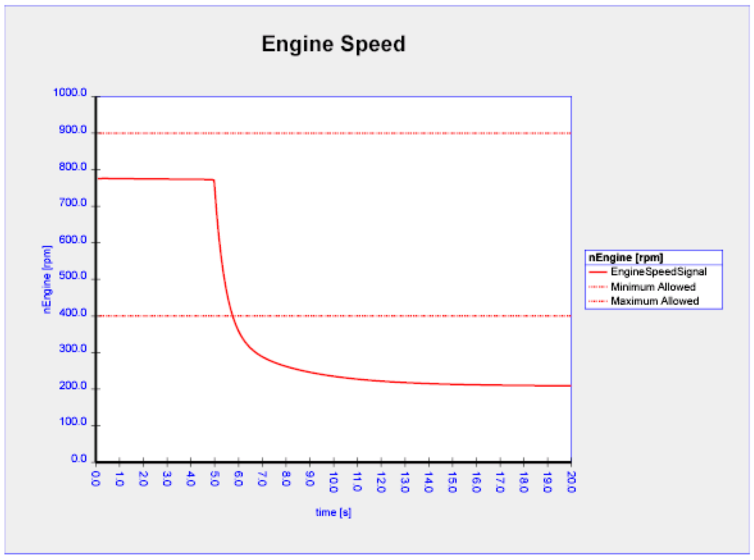
Ignition switched on and engine started

Idle Controller relaxation time : 5

Air Condition Switched On

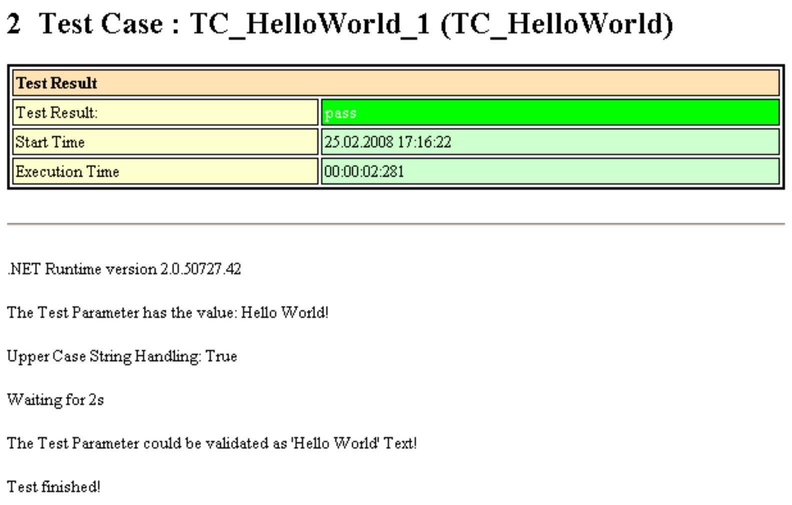
Data Acquisition finished

The engine speed does not recover from the sharp drop caused by the air conditioning being switched on; the (model-specific) speed after activation of around 200 rpm corresponds to a stalled engine. It is assigned the verdict “fail”.



- In the overview, click the test case "TC_HelloWorld_1".

  The test result is "pass" because the value of the test parameter "A Test Parameter" is identical to the predefined string "Hello World".



2 Test Case : TC_HelloWorld_1 (TC_HelloWorld)

Test Result	
Test Result:	pass
Start Time	25.02.2008 17:16:22
Execution Time	00:00:02:281

.NET Runtime version 2.0.50727.42

The Test Parameter has the value: Hello World!

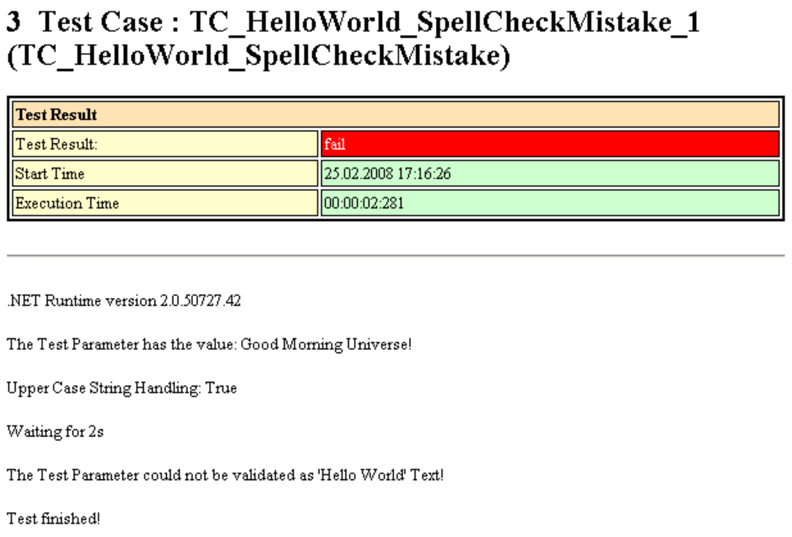
Upper Case String Handling: True

Waiting for 2s

The Test Parameter could be validated as 'Hello World' Text!

Test finished!

With the test case "TC_HelloWorld_SpellCheckMistake_1", "A Test Parameter" contains the string "Good Morning Universe!" which means the comparison ends with the result "fail".



3 Test Case : TC_HelloWorld_SpellCheckMistake_1 (TC_HelloWorld_SpellCheckMistake)

Test Result	
Test Result:	fail
Start Time	25.02.2008 17:16:26
Execution Time	00:00:02:281

.NET Runtime version 2.0.50727.42

The Test Parameter has the value: Good Morning Universe!

Upper Case String Handling: True

Waiting for 2s

The Test Parameter could not be validated as 'Hello World' Text!

Test finished!

6.3 Creating a Test Project

The aim of this section is to recreate the LABCAR-AUTOMATION project which has just been executed in the Test Manager.

You will carry out the following steps:

- "To launch Test Manager" on page 211
- "To set paths" on page 211
- "To create a new LABCAR-AUTOMATION project" on page 213
- "To create further UuT Groups and hierarchies" on page 215
- "To assign UuTs to a UuT Group" on page 216
- "To assign functionality" on page 218
- "To duplicate a test case" on page 221
- "To modify the parameters of a test case" on page 222
- "To extend and parameterize the dynamic array" on page 222
- "To parameterize user records" on page 225
- "To create a new test sequence" on page 226
- "To add functionality to the sequence" on page 226
- "To disable a sequence's test case" on page 229
- "To open the project in the Test Handler" on page 229
- "To create a new test run configuration" on page 230
- "To edit a test run configuration" on page 231
- "To execute a test" on page 232

To launch Test Manager

- In the Windows Start menu, select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2.3** → **Test Manager**.
- Test Manager is launched.

To ensure that the various files and directories for the tutorial project are actually found once installation has been completed, the correct paths have to be set.

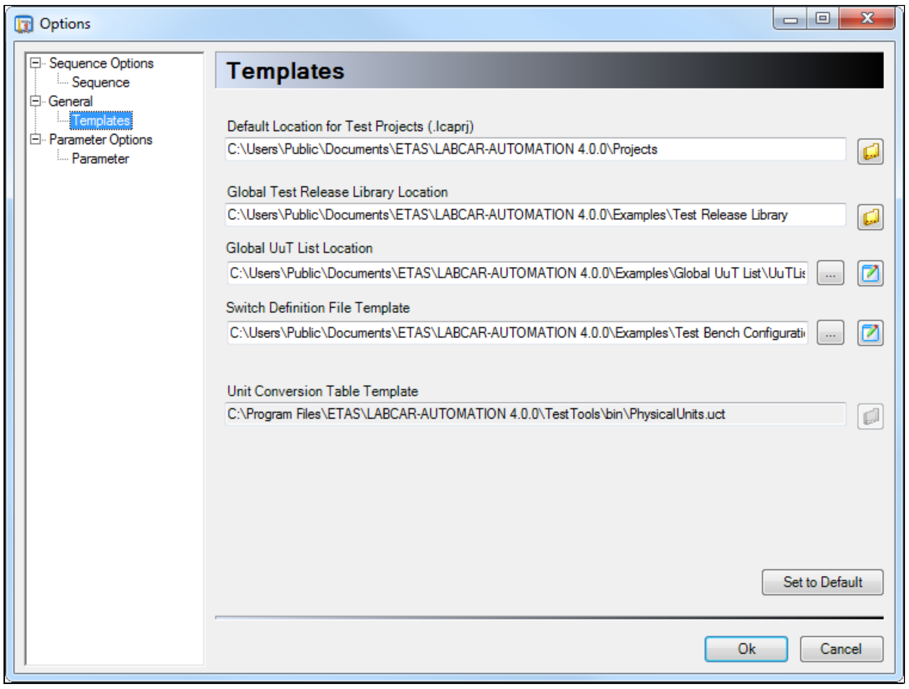
If you have not yet done this, proceed as follows.

To set paths

- Select **Tools** → **Options**.
  The "Options" window opens.

- Select "General → Templates".

  This window shows the paths of all important directories and files.



Templates

Default Location for Test Projects (.lcaprj)
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Projects

Global Test Release Library Location
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Release Library

Global UuT List Location
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Global UuT List\UuTLis

Switch Definition File Template
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Bench Configurati

Unit Conversion Table Template
C:\Program Files\ETAS\LABCAR-AUTOMATION 4.0.0\TestTools\bin\PhysicalUnits.uct

- Select the following settings for the tasks in this tutorial:

Note

These are the default paths in a typical LABCAR-AUTOMATION installation. If you selected a different data directory during installation, you must specify the corresponding paths.

- Default Location for Test Projects:

  `<Drive>:\Users\Public\Documents \ETAS\LABCAR-AUTOMATION 4.2.3\Projects`
- Global Test Release Library Location:

  `<Drive>:\Users\Public\Documents \ETAS\LABCAR-AUTOMATION 4.2.3\Examples \Test Release Library`
- Global UuT List Location:

  `<Drive>:\Users\Public\Documents\ ETAS\LABCAR-AUTOMATION 4.2.3\Examples \Global UuT List\UuTList.dbu`

- Switch Definition File Template:

  `<Drive>:\Users\Public\Documents\ETAS\`
  `LABCAR-AUTOMATION 4.2.3\Examples\Test Bench`
  `Configurations\SwitchDefinitionTable.lcasdt`

Note

These values are saved in the Windows Registry and are thus valid for all projects run on the particular computer.

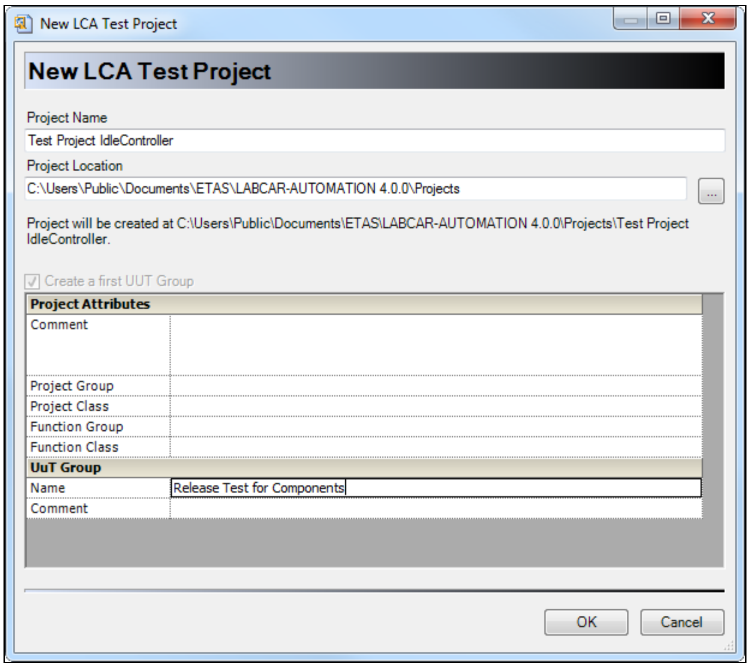
To create a new LABCAR-AUTOMATION project

- In the main menu of Test Manager select **File** → **New Project** .

or

- Click the **New Project** button in the toolbar.

  The "New LCA Test Project" window opens.
- Under "Project Name" enter the project name "Test Project IdleController".
- If necessary, change the preset project directory (see above).
- As a UuT Group is always created with a new project, you must specify a name for this under "UuT Group/Name" ("Release Tests for Components").

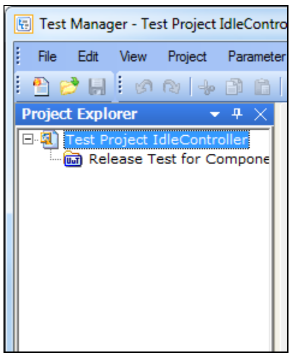


Project Name
Test Project IdleController
Project Location
C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Projects
Project will be created at C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Projects\Test Project IdleController.
Create a first UUT Group [✓]

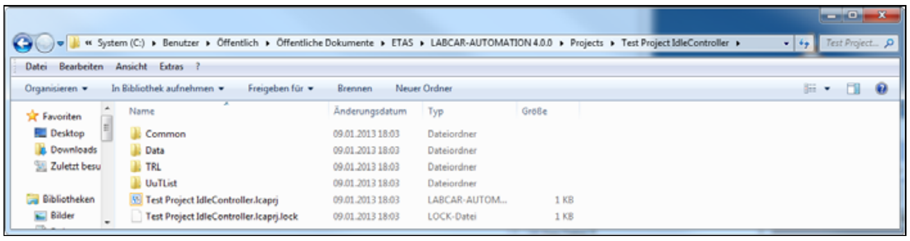
Project Attributes	
Comment	
Project Group	
Project Class	
Function Group	
Function Class	
UuT Group	
Name	Release Test for Components
Comment	

- Click **OK**.

  The new project is created and a UuT Group added.



All folders and files for this project are created at the same time in the above-named directory.

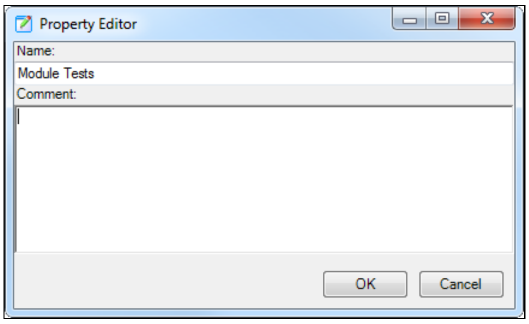


Name	Änderungsdatum	Typ	Größe
Common	09.01.2013 18:03	Dateiordner	
Data	09.01.2013 18:03	Dateiordner	
TRL	09.01.2013 18:03	Dateiordner	
UuTList	09.01.2013 18:03	Dateiordner	
Test Project IdleController.lcaprj	09.01.2013 18:03	LABCAR-AUTOM...	1 KB
Test Project IdleController.lcaprj.lock	09.01.2013 18:03	LOCK-Datei	1 KB

The project has now been created – below, you will create UuT Groups in the project and assign UuTs to them.

To create further UuT Groups and hierarchies

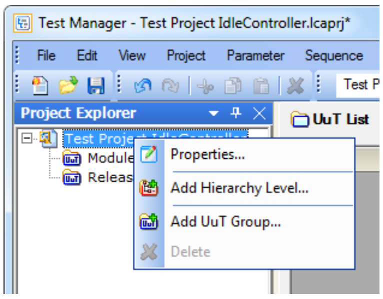
- Select the test project in the Project Explorer.
- Right-click and select **Add UuT Group** from the shortcut menu.
- In the following window, enter the name for the UuT Group to be created ("Module Tests").



Property Editor
Name:
Module Tests
Comment:

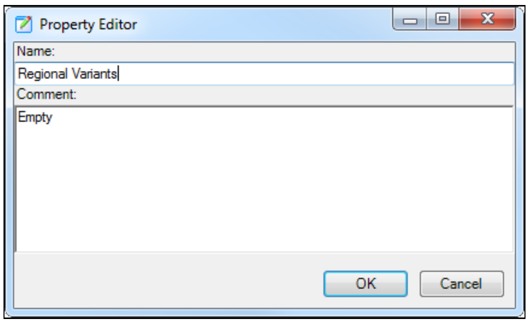
- Click **OK**.

  The UuT Group "Module Tests" is created.
- Select **Add Hierarchy Level** from the shortcut menu.



Test Manager - Test Project IdleController.lcaprj*

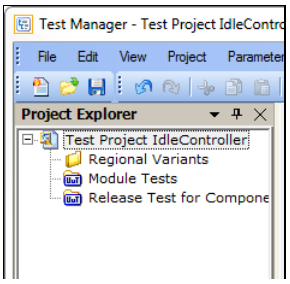
- In the following window, enter the name for the hierarchy level to be created ("Regional Variants").



Property Editor
Name:
Regional Variants
Comment:
Empty

- Click **OK**.

  A new hierarchy level “Regional Variants” is created.


Test Manager - Test Project IdleContro
File Edit View Project Paramete
Project Explorer
Test Project IdleController
Regional Variants
Module Tests
Release Test for Compone

- Below this hierarchy level, create two further UuT Groups “Europe” and “US”.
- To do this, right-click the “Regional Variants” folder and select **Add UuT Group** from the shortcut menu.
- Save the project with **Save**.

You have now created several UuT Groups and a hierarchy level in the Project Explorer.

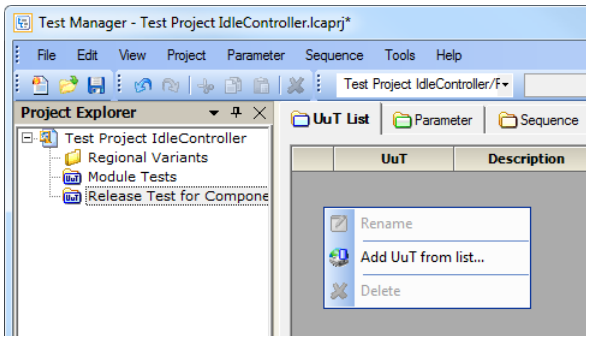
In the next step you will assign the UuT Group “Release Tests for Components” various UuTs which come from the “Global UuT List” supplied with the tutorial project.

Note

For general information on how to deal with UuT Lists, please refer to the section "The UuT List Editor" on page 186.

To assign UuTs to a UuT Group

- In the Project Explorer, select the UuT Group “Release Tests for Components”.
- Right-click in the “UuT List” tab.
- From the shortcut menu, select **Add UuT from list**.


Test Manager - Test Project IdleController.lcaprj*
File Edit View Project Parameter Sequence Tools Help
Test Project IdleController/F
Project Explorer
Test Project IdleController
Regional Variants
Module Tests
Release Test for Compone
UuT List
Parameter
Sequence
UuT
Description
Rename
Add UuT from list...
Delete

  The “Add from Global UuT List” window opens.

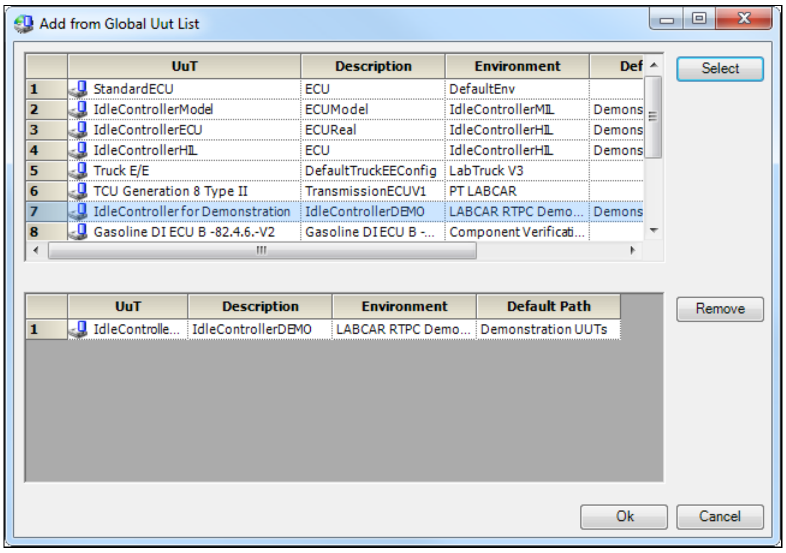
- Select the UuT to be added, “IdleController for Demonstration”, in the top part of the window.

- Click **Select**.

or

- Double-click the UuT required.

  The selected UuT is displayed in the lower part of the window (= the list of UuTs to be added).

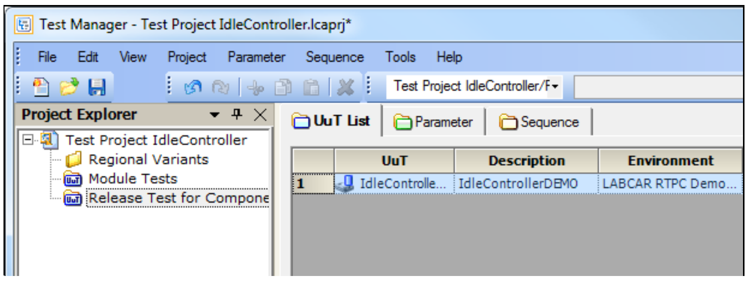


	UuT	Description	Environment	Def
1	StandardECU	ECU	DefaultEnv	
2	IdleControllerModel	ECUModel	IdleControllerMIL	Demons
3	IdleControllerECU	ECUReal	IdleControllerHIL	Demons
4	IdleControllerHIL	ECU	IdleControllerHIL	Demons
5	Truck E/E	DefaultTruckEEConfig	LabTruck V3	
6	TCU Generation 8 Type II	TransmissionECUV1	PT LABCAR	
7	IdleController for Demonstration	IdleControllerDEMO	LABCAR RTPC Demo...	Demons
8	Gasoline DI ECU B -82.4.6.-V2	Gasoline DI ECU B -...	Component Verificati...	

	UuT	Description	Environment	Default Path
1	IdleControlle...	IdleControllerDEMO	LABCAR RTPC Demo...	Demonstration UUTs

- Click **OK**.

  The UuT "IdleController for Demonstration" is added.



	UuT	Description	Environment
1	IdleControlle...	IdleControllerDEMO	LABCAR RTPC Demo...

- Add more UuTs. To do this, select the following UuTs from the "Add from Global UuT List" window keeping the <CTRL> key pressed down:
  - StandardECU
  - TCU Generation 8 Type II
  - Gasoline DI ECU B -82.4.6.-V2
  - Diesel Control 17 - Platform A V5.6
- Click **Select** and **OK**.

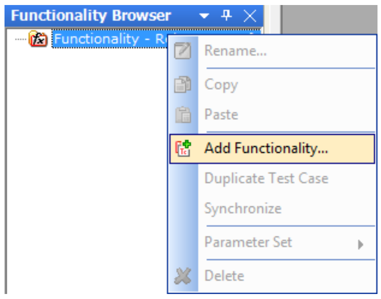
  The selected UuTs are added to the UuT Group "Release Tests for Components".

- Save the project with **Save**.

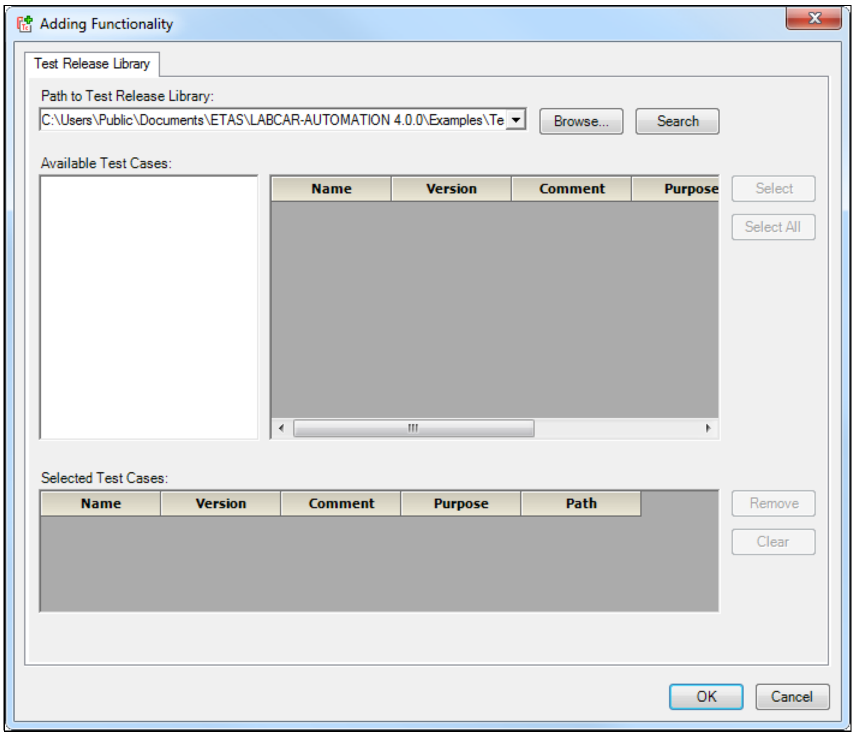
You have now added UuTs to the UuT Group “Release Tests for Components”. In the next step, you will add test functionality to this UuT Group.

To assign functionality

- In the Project Explorer select the UuT Group (“Release Tests for Components”) to which you want to add functionality (Test Cases).
- Right-click in the Functionality Browser.
- From the shortcut menu, select **Add Functionality**.



The “Adding Functionality” window opens.



Note

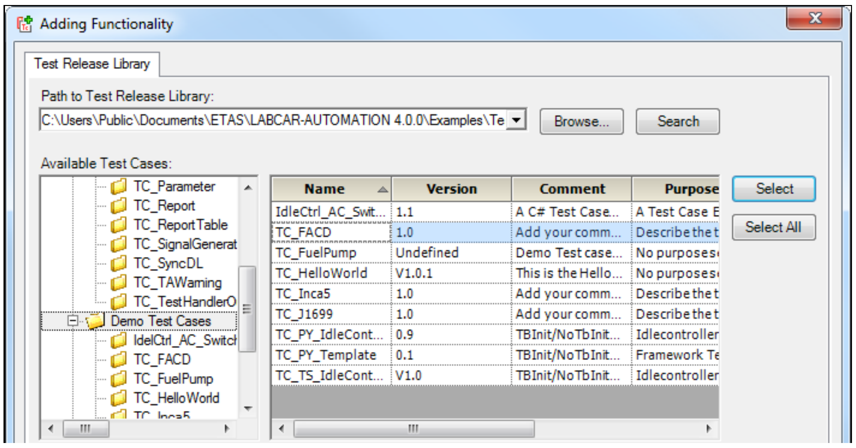
If the path to the Global Test Release Library or the Global Test Case Collection Library was not specified, this window might not open. In this case, check whether the paths have been set (see "To set paths" on page 211).

The path shown under “Path to Test Release Library” is the default path you set earlier (see "To set paths" on page 211).
You can select a different directory using **Browse**.

- Click **Search**.

  The Test Release Library is searched and the available test cases shown in a list.

- Select the test case to be added (here: "IdleCtrl_AC_SwitchingTest").



Path to Test Release Library: C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Te

Available Test Cases:

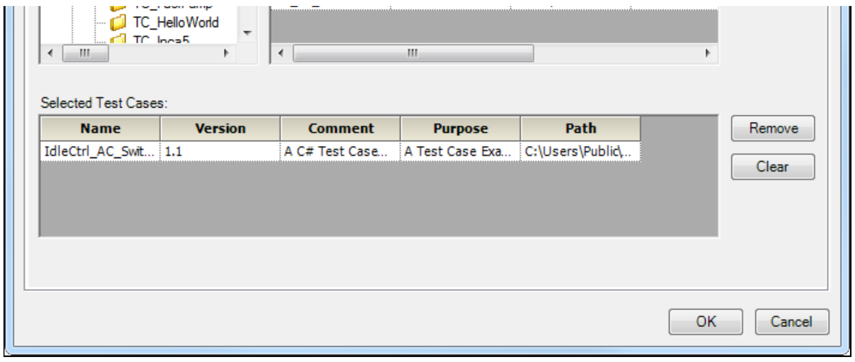
Name	Version	Comment	Purpose
IdleCtrl_AC_Swit...	1.1	A C# Test Case...	A Test Case E
TC_FACD	1.0	Add your comm...	Describe the t
TC_FuelPump	Undefined	Demo Test case...	No purposes
TC_HelloWorld	V1.0.1	This is the Hello...	No purposes
TC_Inca5	1.0	Add your comm...	Describe the t
TC_J1699	1.0	Add your comm...	Describe the t
TC_PY_IdleCont...	0.9	TBInit/NoTbInit...	Idlecontroller
TC_PY_Template	0.1	TBInit/NoTbInit...	Framework Te
TC_TS_IdleCont...	V1.0	TBInit/NoTbInit...	Idlecontroller

- Click **Select**.

or

- Double-click the test case to be added.

  The test case is added to the "Selected Test Cases" list.

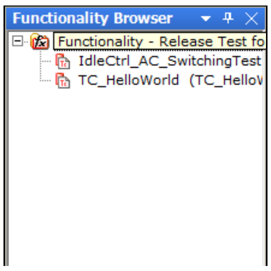


Selected Test Cases:

Name	Version	Comment	Purpose	Path
IdleCtrl_AC_Swit...	1.1	A C# Test Case...	A Test Case Exa...	C:\Users\Public\...

- Add another test case ("TC_HelloWorld") to the list in exactly the same way.
- Click **OK**.

  The selected test case is added in the Functionality Browser.

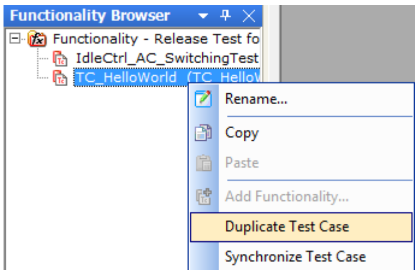


- Save the project with **Save**.

Below, you will duplicate the test case "TC_HelloWorld" to reuse it under another name.

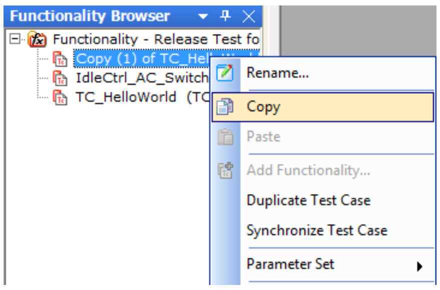
To duplicate a test case

- Select the test case “TC_HelloWorld” from the Functionality Browser.
- Select **Duplicate Test Case** from the shortcut menu.



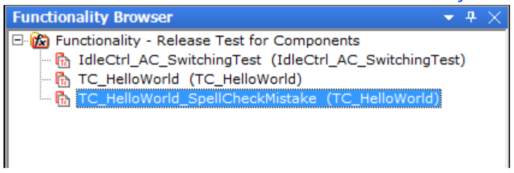
The test case “Copy (1) of TC_HelloWorld” is created.

- Select this test case and right-click.
- From the shortcut menu, select **Rename**.



- Rename the test case “TC_HelloWorld_SpellCheckMistake”.

You have now added all the functionality.


Functionality Browser
Functionality - Release Test for Components
IdleCtrl_AC_SwitchingTest (IdleCtrl_AC_SwitchingTest)
TC_HelloWorld (TC_HelloWorld)
TC_HelloWorld_SpellCheckMistake (TC_HelloWorld)

- Save the project with **Save**.

Once the necessary functionality has been assigned, we can now concentrate on parameterizing the test.

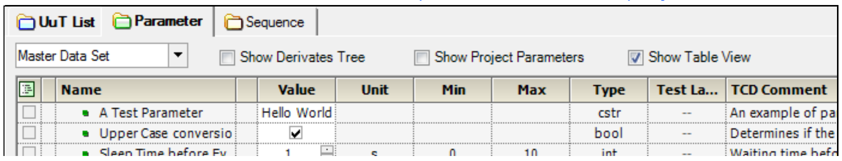
Test cases are assigned a default parameterization by the Test Case Developer; if necessary, this is adapted by the Test Parameter Manager (TPM).

To view the parameters of a test case

- In the Project Explorer, select the UuT Group “Release Tests for Components”.
- Select the test case “TC_HelloWorld” in the Functionality Browser.

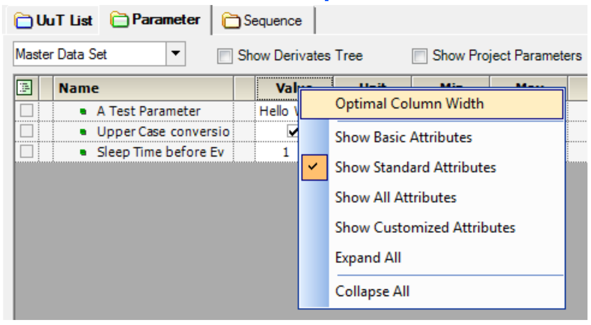
- Activate the “Parameter” tab.

  The test case parameters are displayed.



Name	Value	Unit	Min	Max	Type	Test La...	TCD Comment
A Test Parameter	Hello World				cstr	--	An example of pa
Upper Case conversio	✓				bool	--	Determines if the
Sleep Time before Ev	1	s	0	10	int	--	Waiting time befo

- If the entries in the table are not legible because the column widths have not been adapted, right-click the table head and select **Optimal Column Width**.

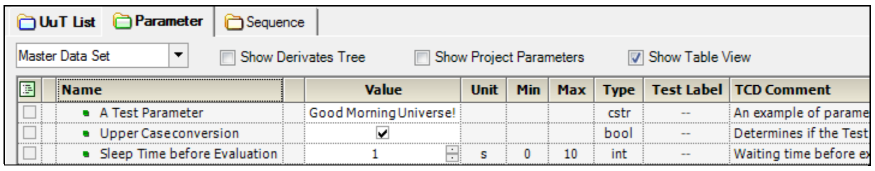


The parameter “A Test Parameter” has the value “Hello World!” for the current test case.

In this test case, the content of this parameter is compared to a predefined string (“Hello World!”) and a corresponding test result (verdict) issued (if they match “pass”, otherwise “fail”).

To modify the parameters of a test case

- Now select the test case “TC_HelloWorld_SpellCheckMistake” from the Functionality Browser.
- Change “A Test Parameter” to “Good Morning Universe!”



Name	Value	Unit	Min	Max	Type	Test Label	TCD Comment
A Test Parameter	Good Morning Universe!				cstr	--	An example of parame
Upper Case conversion	✓				bool	--	Determines if the Test
Sleep Time before Evaluation	1	s	0	10	int	--	Waiting time before ex

Running this test case will result in the verdict “fail”.

- Save the project with **Save**.

To check the test case for the idle controller with different variants regarding the torque or the permissible speed range, extend the dynamic array.

To extend and parameterize the dynamic array

- Now select the test case “IdleCtrl_AC_SwitchingTest” from the Functionality Browser.

- Select the "Parameter" tab.
- If necessary, disable the option "Show Table View" (above the table).

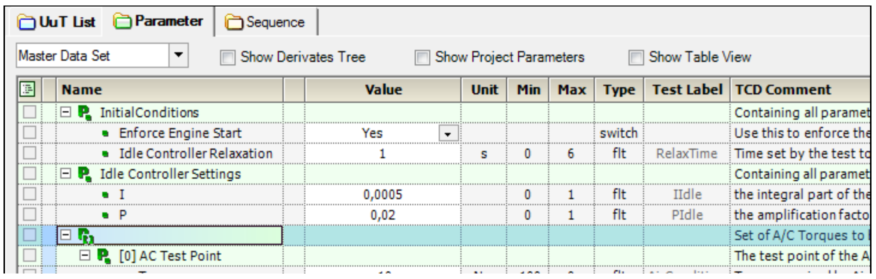
All parameters of the selected test case are displayed in the list:

– The "Initial Conditions" user record which contains two further parameters.

– The "Idle Controller Settings" user record which also contains two further parameters.

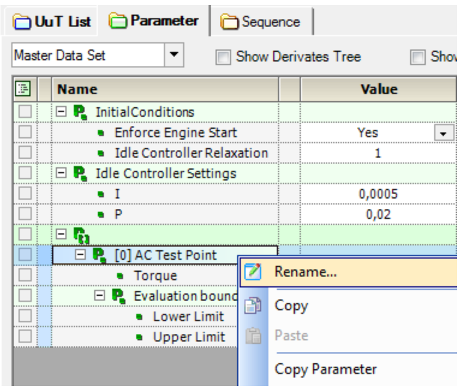
– A dynamic array which (so far) consists of one user record "AC Test Point".

This user record contains the parameter "Torque" and the user record "Evaluation boundaries Limits", the "Lower Limit" and "Upper Limit" parameters of which define the lower and upper limit of the permissible speed range.



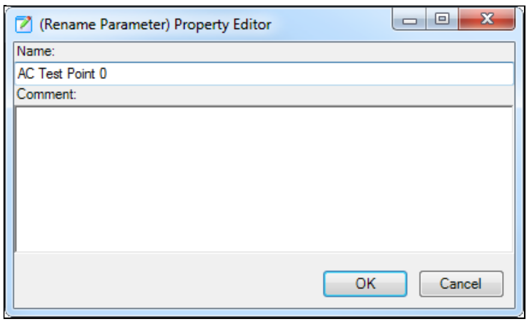
Name	Value	Unit	Min	Max	Type	Test Label	TCD Comment
⊟ InitialConditions							Containing all paramet
Enforce Engine Start	Yes				switch		Use this to enforce the
Idle Controller Relaxation	1	s	0	6	flt	RelaxTime	Time set by the test to
⊟ Idle Controller Settings							Containing all paramet
I	0,0005		0	1	flt	IIdle	the integral part of the
P	0,02		0	1	flt	PIdle	the amplification facto
⊟							Set of A/C Torques to
⊟ [0] AC Test Point							The test point of the A

- To assign further elements to the dynamic array, proceed as follows.
- Select the user record "AC Test Point".
- Right-click and select **Rename** from the shortcut menu.

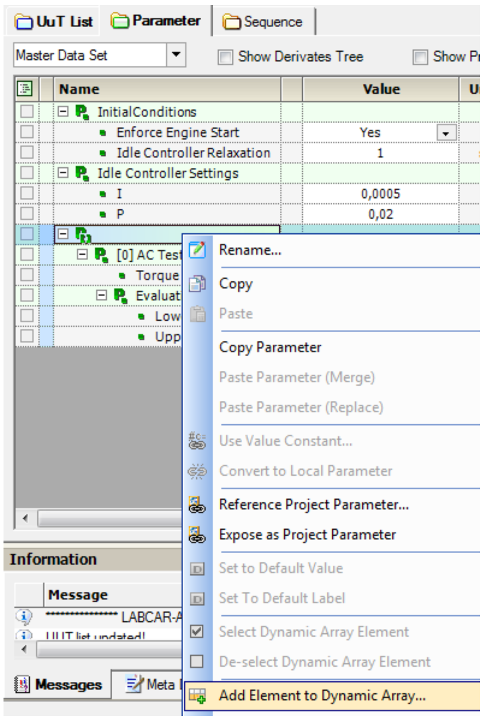


Name	Value
⊟ InitialConditions	
Enforce Engine Start	Yes
Idle Controller Relaxation	1
⊟ Idle Controller Settings	
I	0,0005
P	0,02
⊟	
⊟ [0] AC Test Point	
Torque	
⊟ Evaluation bound	
Lower Limit	
Upper Limit	

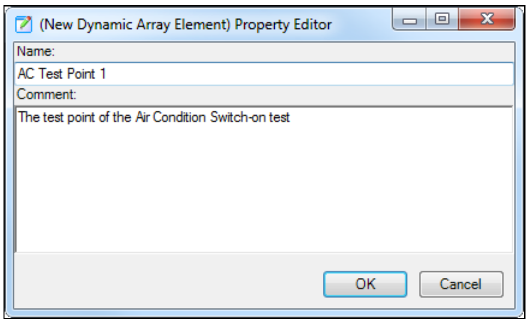
- Rename the user record “AC Test Point 0”.



- Right-click and select **Add Element to Dynamic Array** from the shortcut menu.



- Assign the new element the name “AC Test Point 1”.


(New Dynamic Array Element) Property Editor
Name:
AC Test Point 1
Comment:
The test point of the Air Condition Switch-on test
OK
Cancel

- Add two further elements in the same way, “AC Test Point 2” and “AC Test Point 3”.

- Save the project with **Save**.

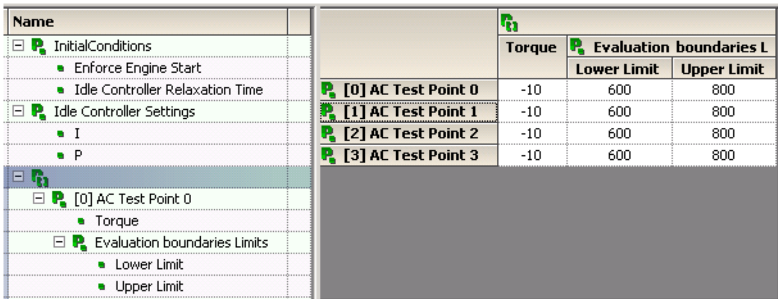
When the test project is run, the elements of the dynamic array are run in succession – this of course takes place with varying parameterization.

To parameterize user records

☑ Show Table View

- If necessary, enable the option "Show Table View" (above the table).

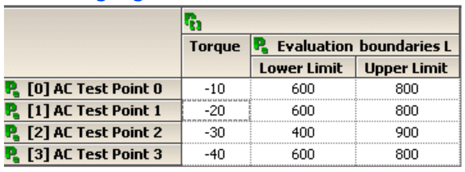
  The list displays all elements of the dynamic array in tabular form.



```
Name
⊟ InitialConditions
    Enforce Engine Start
    Idle Controller Relaxation Time
⊟ Idle Controller Settings
    I
    P
⊟
  ⊟ [0] AC Test Point 0
      Torque
    ⊟ Evaluation boundaries Limits
        Lower Limit
        Upper Limit
```

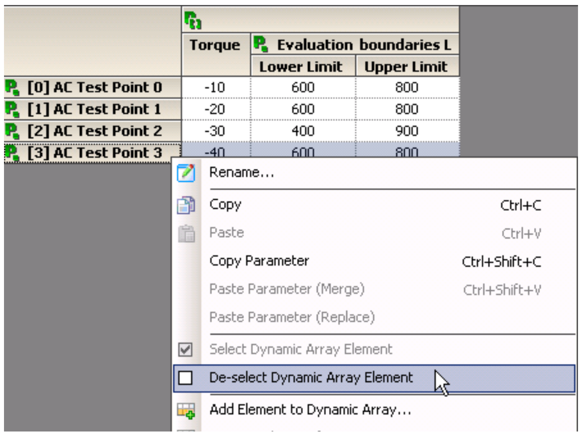
	Torque	Evaluation boundaries L	
		Lower Limit	Upper Limit
[0] AC Test Point 0	-10	600	800
[1] AC Test Point 1	-10	600	800
[2] AC Test Point 2	-10	600	800
[3] AC Test Point 3	-10	600	800

- Now change the parameterization as shown in the following figure.



	Torque	Evaluation boundaries L	
		Lower Limit	Upper Limit
[0] AC Test Point 0	-10	600	800
[1] AC Test Point 1	-20	600	800
[2] AC Test Point 2	-30	400	900
[3] AC Test Point 3	-40	600	800

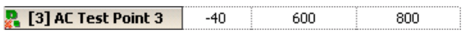
- Select the user record "AC Test Point 3".
- Right-click and select "Deselect Dynamic Array Element" from the shortcut menu.



	Torque	Evaluation boundaries L	
		Lower Limit	Upper Limit
[0] AC Test Point 0	-10	600	800
[1] AC Test Point 1	-20	600	800
[2] AC Test Point 2	-30	400	900
[3] AC Test Point 3	-40	600	800

Rename...
Copy Ctrl+C
Paste Ctrl+V
Copy Parameter Ctrl+Shift+C
Paste Parameter (Merge) Ctrl+Shift+V
Paste Parameter (Replace)
Select Dynamic Array Element
De-select Dynamic Array Element
Add Element to Dynamic Array...

The element of the dynamic array is retained, but is not run during the test (indicated by the red cross).



[3] AC Test Point 3	-40	600	800

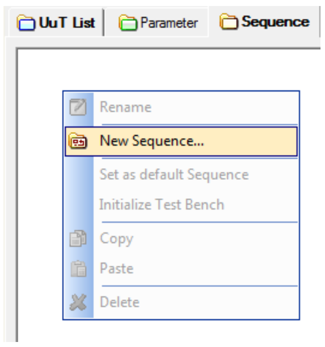
- Save the project with **Save**.

The "IdleCtrl_AC_SwitchingTest" test case now been parameterized.

Once the test case has been parameterized, you now have to create test sequences which define the order in which the test cases are run and the test bench initialization.

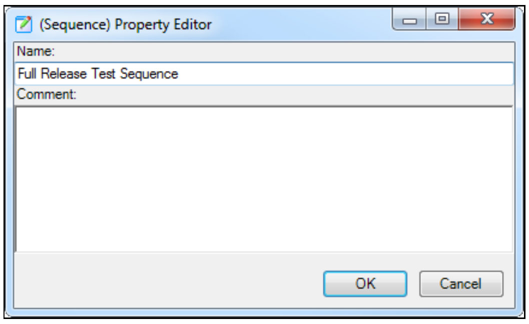
To create a new test sequence

- Activate the "Sequence" tab.
- Select **New Sequence** from the shortcut menu of the list of sequences (on the left).



The (Sequence) Property Editor opens.

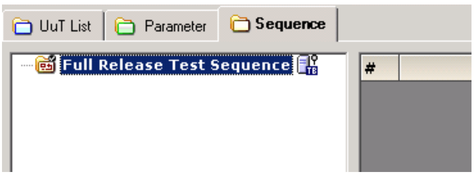
- Enter a name ("Full Release Test Sequence") and, if desired, a comment.



Name:
Full Release Test Sequence
Comment:

- Click **OK**.

The new sequence is created and automatically becomes the default sequence (the default sequence is shown in bold).



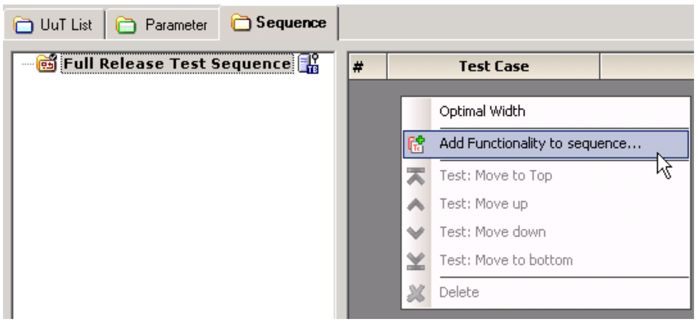
- The sequence is automatically assigned a test bench initialization.

Now you will select test functionality from the project for this sequence.

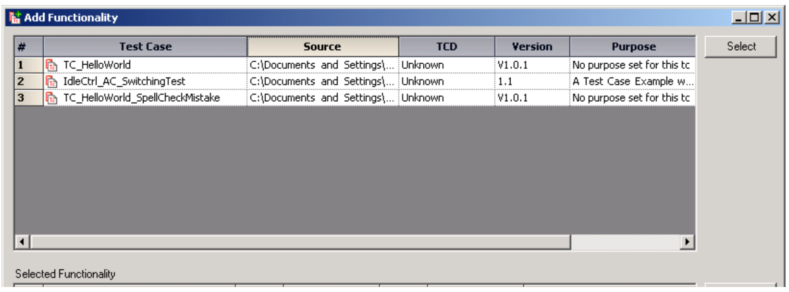
To add functionality to the sequence

- Select the relevant sequence.
- Right-click the list of test cases.

- Select **Add Functionality to Sequence** from the shortcut menu.



The "Add Functionality" window opens.

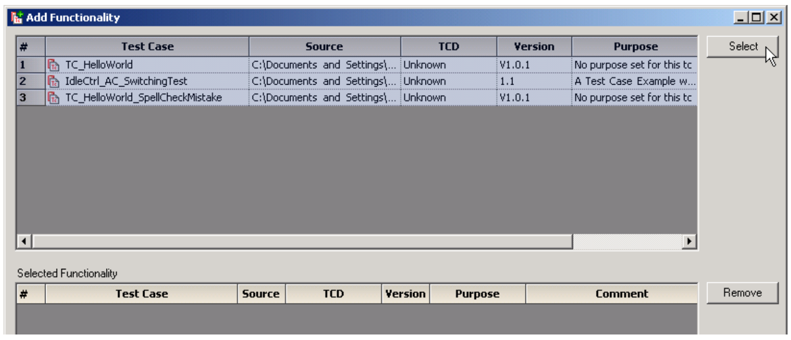


#	Test Case	Source	TCD	Version	Purpose
1	TC_HelloWorld	C:\Documents and Settings\...	Unknown	V1.0.1	No purpose set for this tc
2	IdleCtrl_AC_SwitchingTest	C:\Documents and Settings\...	Unknown	1.1	A Test Case Example w...
3	TC_HelloWorld_SpellCheckMistake	C:\Documents and Settings\...	Unknown	V1.0.1	No purpose set for this tc

- Select the test cases on offer (in the top part of the window) and click **Select**.

or

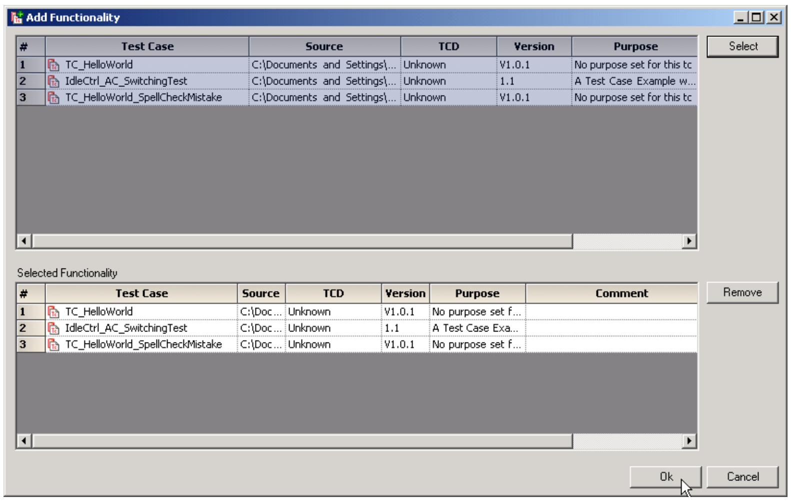
- Double-click the test cases one after the other.



#	Test Case	Source	TCD	Version	Purpose
1	TC_HelloWorld	C:\Documents and Settings\...	Unknown	V1.0.1	No purpose set for this tc
2	IdleCtrl_AC_SwitchingTest	C:\Documents and Settings\...	Unknown	1.1	A Test Case Example w...
3	TC_HelloWorld_SpellCheckMistake	C:\Documents and Settings\...	Unknown	V1.0.1	No purpose set for this tc

#	Test Case	Source	TCD	Version	Purpose	Comment

The test cases are added to the “Selected Functionality” list (in the bottom part of the window).

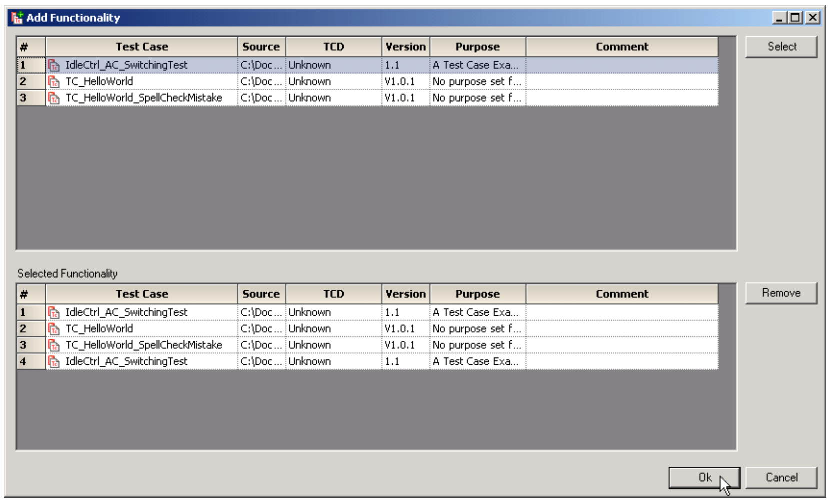


#	Test Case	Source	TCD	Version	Purpose
1	TC_HelloWorld	C:\Documents and Settings\...	Unknown	V1.0.1	No purpose set for this tc
2	IdleCtrl_AC_SwitchingTest	C:\Documents and Settings\...	Unknown	1.1	A Test Case Example w...
3	TC_HelloWorld_SpellCheckMistake	C:\Documents and Settings\...	Unknown	V1.0.1	No purpose set for this tc

Selected Functionality

#	Test Case	Source	TCD	Version	Purpose	Comment
1	TC_HelloWorld	C:\Doc...	Unknown	V1.0.1	No purpose set f...	
2	IdleCtrl_AC_SwitchingTest	C:\Doc...	Unknown	1.1	A Test Case Exa...	
3	TC_HelloWorld_SpellCheckMistake	C:\Doc...	Unknown	V1.0.1	No purpose set f...	

- From the top list, select the test case “IdleCtrl_AC_SwitchingTest” again and add it to the list of selected test functionality using Select.



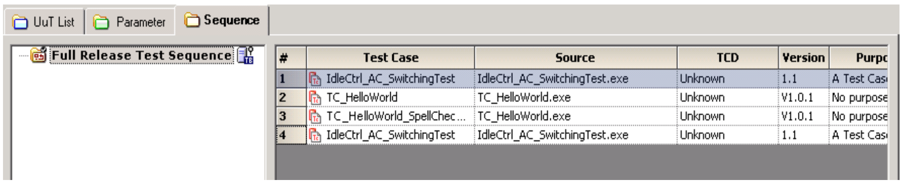
#	Test Case	Source	TCD	Version	Purpose	Comment
1	IdleCtrl_AC_SwitchingTest	C:\Doc...	Unknown	1.1	A Test Case Exa...	
2	TC_HelloWorld	C:\Doc...	Unknown	V1.0.1	No purpose set f...	
3	TC_HelloWorld_SpellCheckMistake	C:\Doc...	Unknown	V1.0.1	No purpose set f...	

Selected Functionality

#	Test Case	Source	TCD	Version	Purpose	Comment
1	IdleCtrl_AC_SwitchingTest	C:\Doc...	Unknown	1.1	A Test Case Exa...	
2	TC_HelloWorld	C:\Doc...	Unknown	V1.0.1	No purpose set f...	
3	TC_HelloWorld_SpellCheckMistake	C:\Doc...	Unknown	V1.0.1	No purpose set f...	
4	IdleCtrl_AC_SwitchingTest	C:\Doc...	Unknown	1.1	A Test Case Exa...	

- Click **OK**.

  The selected test cases are added to the selected sequence.



#	Test Case	Source	TCD	Version	Purpc
1	IdleCtrl_AC_SwitchingTest	IdleCtrl_AC_SwitchingTest.exe	Unknown	1.1	A Test Cas
2	TC_HelloWorld	TC_HelloWorld.exe	Unknown	V1.0.1	No purpose
3	TC_HelloWorld_SpellChec...	TC_HelloWorld.exe	Unknown	V1.0.1	No purpose
4	IdleCtrl_AC_SwitchingTest	IdleCtrl_AC_SwitchingTest.exe	Unknown	1.1	A Test Cas

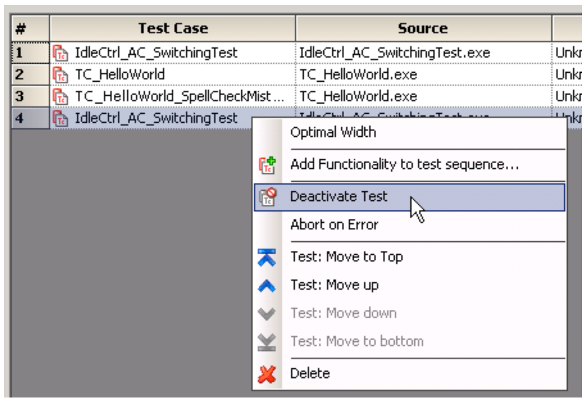
Note

You can also add the desired functionality to the “Sequence” tab from the Functionality Browser using Drag & Drop.

The Test Manager can disable a sequence's test case, i.e. the test case is officially part of the sequence, but is not run.

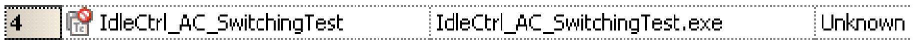
To disable a sequence's test case

- Select the second entity of the test case "IdleCtrl_AC_SwitchingTest".
- Right-click and select **Deactivate Test** from the shortcut menu.



#	Test Case	Source	
1	IdleCtrl_AC_SwitchingTest	IdleCtrl_AC_SwitchingTest.exe	Unk
2	TC_HelloWorld	TC_HelloWorld.exe	Unk
3	TC_HelloWorld_SpellCheckMist...	TC_HelloWorld.exe	Unk
4	IdleCtrl_AC_SwitchingTest	IdleCtrl_AC_SwitchingTest.exe	Unk

The test case is disabled.



4	IdleCtrl_AC_SwitchingTest	IdleCtrl_AC_SwitchingTest.exe	Unknown

- Save the project with **Save**.
- Exit Test Manager with **File** → **Exit**.

You have now created the test project, which can be opened and run in the Test Handler.

To open the project in the Test Handler

- In the Windows Start menu, select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2.3** → **Test Handler**.

  Test Handler is launched.
- Select **File** → **Open Project**.
- Select the project file of the project just created from the file selection window.

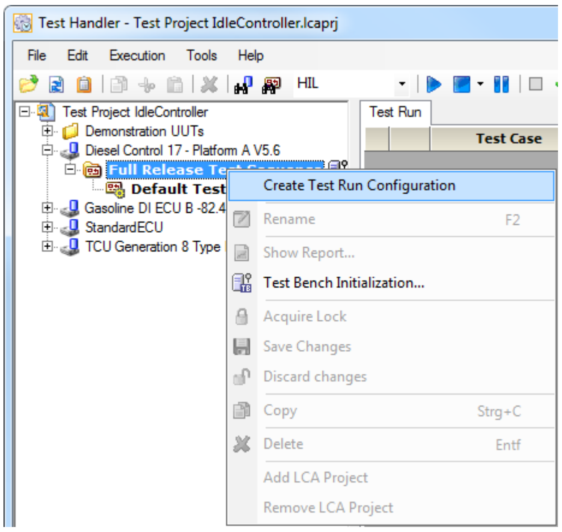
  The project is opened in Test Handler.

The "Full Release Test Sequence" contains the "Default Test Run Configuration" created automatically which in turn contains all test cases assigned to the UuT each of which is run once.

If you want to run variants (in terms of the order or number of times executed) when processing the test cases, create a new test run configuration.

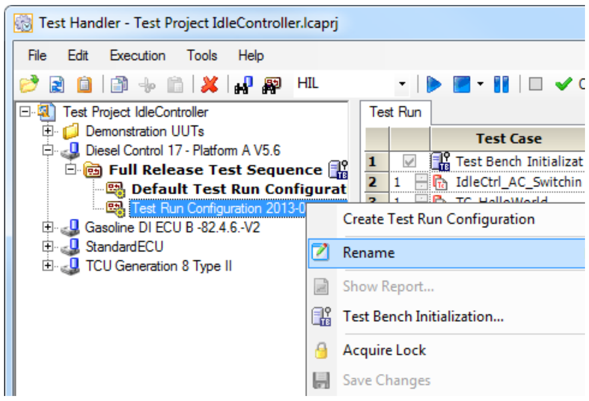
To create a new test run configuration

- Select the test sequence to which you want to add a test run configuration.
- Right-click and select **Create Test Run Configuration** from the shortcut menu.



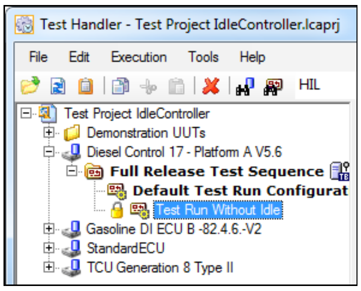
A new test run configuration is created with date and time. It contains the same test cases as the default test run configuration.

- From the shortcut menu, select **Rename**.


Test Handler - Test Project IdleController.lcaprj
File
Edit
Execution
Tools
Help
HIL
Test Project IdleController
Demonstration UUTs
Diesel Control 17 - Platform A V5.6
Full Release Test Sequence
Default Test Run Configurat
Test Run Configuration 2013-0
Gasoline DI ECU B -82.4.6.-V2
StandardECU
TCU Generation 8 Type II
Test Run
Test Case
Test Bench Initializat
IdleCtrl_AC_Switchin
Create Test Run Configuration
Rename
Show Report...
Test Bench Initialization...
Acquire Lock
Save Changes

The name can now be edited; a small padlock symbol is added to the front of the name. This shows that this test run configuration is now locked for you.

- Rename the test run configuration “Test Run Without Idle”.



Test Handler - Test Project IdleController.lcaprj

File Edit Execution Tools Help

HIL

- Test Project IdleController
  - Demonstration UUTs
  - Diesel Control 17 - Platform A V5.6
    - **Full Release Test Sequence**
      - **Default Test Run Configurat**
      - Test Run Without Idle
  - Gasoline DI ECU B -82.4.6.-V2
  - StandardECU
  - TCU Generation 8 Type II

- Right-click and select **Save Changes** from the shortcut menu.

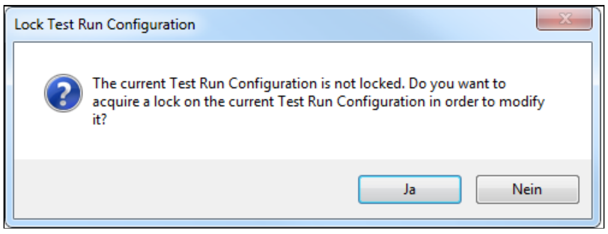
  The test run configuration lock is cancelled.

Now, the newly created test run configuration is to be edited so that only the test case “TC_HelloWord” and its successor “TC_HelloWord_SpellCheckMistake” are checked.

To edit a test run configuration

- Click the list of test cases.

  The following message is displayed.

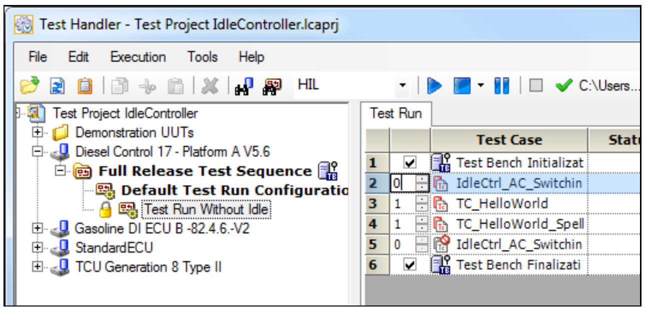


Lock Test Run Configuration

The current Test Run Configuration is not locked. Do you want to acquire a lock on the current Test Run Configuration in order to modify it?

Ja Nein

- Click **Yes**.
- Click the “1” in front of the test case “IdleCtrl_AC_SwitchingTest”.
- Set the value to “0” (by entering it directly or using the spin buttons).



Test Handler - Test Project IdleController.lcaprj

File Edit Execution Tools Help

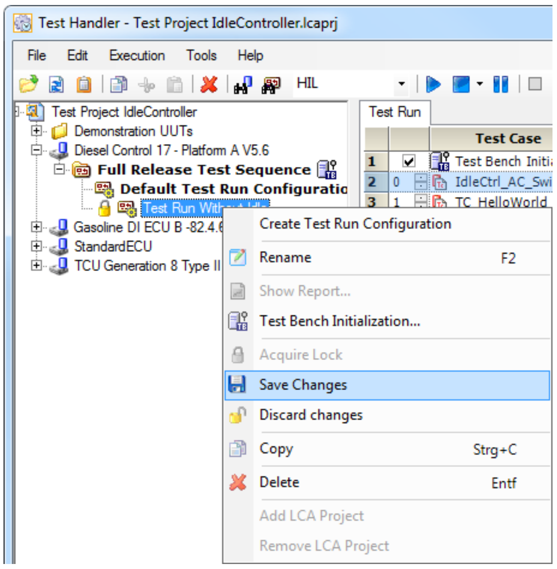
HIL C:\Users...

- Test Project IdleController
  - Demonstration UUTs
  - Diesel Control 17 - Platform A V5.6
    - **Full Release Test Sequence**
      - **Default Test Run Configuratio**
      - Test Run Without Idle
  - Gasoline DI ECU B -82.4.6.-V2
  - StandardECU
  - TCU Generation 8 Type II

Test Run

		Test Case	Stat
1	✓	Test Bench Initializat	
2	0	IdleCtrl_AC_Switchin	
3	1	TC_HelloWorld	
4	1	TC_HelloWorld_Spell	
5	0	IdleCtrl_AC_Switchin	
6	✓	Test Bench Finalizati	

- Select **Save Changes** from the shortcut menu to save the changes and re-release the test run configuration.



Test Handler - Test Project IdleController.lcaprj

File Edit Execution Tools Help

HIL

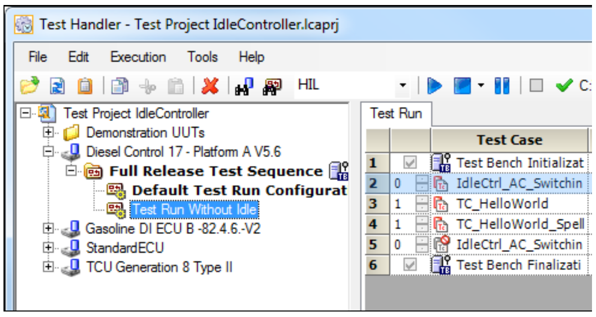
Test Project IdleController
- Demonstration UUTs
- Diesel Control 17 - Platform A V5.6
  - **Full Release Test Sequence**
    - **Default Test Run Configuratio**
      - Test Run With
- Gasoline DI ECU B -82.4.6
- StandardECU
- TCU Generation 8 Type II

Test Run

		Test Case
1	✓	Test Bench Initi
2	0	IdleCtrl_AC_Swi
3	1	TC_HelloWorld

Create Test Run Configuration
Rename F2
Show Report...
Test Bench Initialization...
Acquire Lock
Save Changes
Discard changes
Copy Strg+C
Delete Entf
Add LCA Project
Remove LCA Project

The test run configuration now no longer contains an IdleController test run.



Test Handler - Test Project IdleController.lcaprj

File Edit Execution Tools Help

HIL

Test Project IdleController
- Demonstration UUTs
- Diesel Control 17 - Platform A V5.6
  - **Full Release Test Sequence**
    - **Default Test Run Configurat**
      - Test Run Without Idle
- Gasoline DI ECU B -82.4.6.-V2
- StandardECU
- TCU Generation 8 Type II

Test Run

		Test Case
1	✓	Test Bench Initializat
2	0	IdleCtrl_AC_Switchin
3	1	TC_HelloWorld
4	1	TC_HelloWorld_Spell
5	0	IdleCtrl_AC_Switchin
6	✓	Test Bench Finalizati

To execute a test

See "Running a Test Project" on page 199.

6.4 Configuring the Test Bench

A range of configuration files are necessary for the successful running of a test. These are supplied with the tutorial project – for more details on these files, refer to the section "Configuration Files and their Editors" on page 147.

This section contains a short introduction on how to view the configuration files used in the tutorial project.

6.4.1 The Test Bench Configuration File

The test bench configuration file is selected by the Test Handler before the test is run (see "To provide a test bench configuration" on page 201).

To launch the Test Bench Configuration File Editor

- In the Windows Start menu, select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2.3** → **Editors** → **Test Bench Configuration File Editor**.

  The Test Bench Configuration File Editor opens.

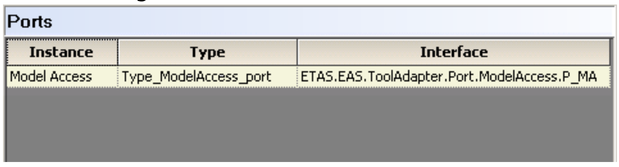
To load the test bench configuration file

- Select **File** → **Open**.

  A file selector window opens.
- Select the file *<Drive>*`:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2.3\Examples\Test Bench Configurations\Demo Test Bench\LCO 5.2\LabCar_RTPCDemo.tbc`.

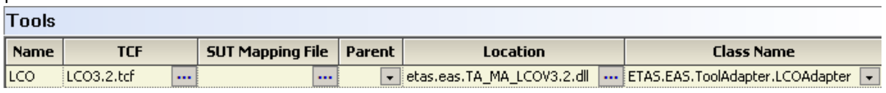
  The selected test bench configuration file is loaded.

The "Model Access" port entity was defined in the "Ports" field – this is the port for accessing the simulation model.


Ports

Instance	Type	Interface
Model Access	Type_ModelAccess_port	ETAS.EAS.ToolAdapter.Port.ModelAccess.P_MA

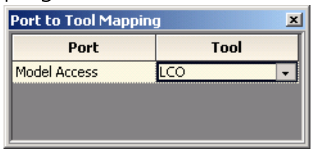
In the file, the "LCO" tool (stands for LABCAR-OPERATOR) which was implemented in the `etas.eas.TA_MA_LCOV3.dll` file, is defined for this abstract port.


Tools

Name	TCF	SUT Mapping File	Parent	Location	Class Name
LCO	LCO3.2.tcf			etas.eas.TA_MA_LCOV3.2.dll	ETAS.EAS.ToolAdapter.LCOAdapter

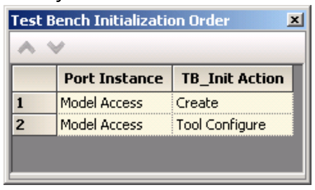
For further information on the tool configuration, please refer to the tool configuration file `lco.tcf` (see "The Tool Configuration File" on page 234). This file is in the following path: *<Drive>*`:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2.3\Examples\Test Bench Configurations\Demo Test Bench\LCO 5.2\LCO.tcf`.

The “Model Access” port is mapped to the “LCO” tool in the “Port to Tool Mapping” field.


Port to Tool Mapping

Port	Tool
Model Access	LCO

The test bench initialization is determined in the “Test Bench Initialization Order” field in which a “Create” and then a “Tool Configure” is carried out for the port entity “Model Access”.


Test Bench Initialization Order

	Port Instance	TB_Init Action
1	Model Access	Create
2	Model Access	Tool Configure

The test bench configuration file is assigned to the project later by the person executing the test (see "To provide a test bench configuration" on page 201).

Select **File** → **Exit** to close the editor.

6.4.2 The Tool Configuration File

The Tool Configuration File `lco.tcf`, defined for the “Model Access” port and its tool “LCO” in the test bench configuration file, contains the configurations for the tool “LABCAR-OPERATOR”.

To start the Tool Configuration File Editor

- In the Windows Start menu, select **All Programs** → **ETAS** → **LABCAR-AUTOMATION 4.2.3** → **Editors** → **Tool Configuration File Editor**.

  The Tool Configuration File Editor opens.

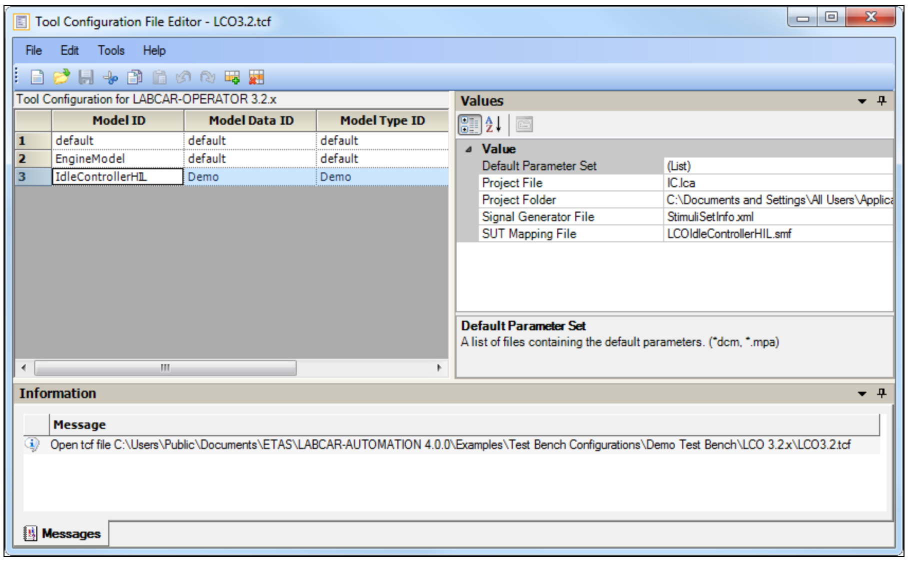
To open the tool configuration file

- Select **File** → **Open**.

  A file selector window opens.

- Select the file <*Drive*>`:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2.3\Examples\Test Bench Configurations\Demo Test Bench\LCO 5.2\LCO.tcf`.

  The selected tool configuration file opens.



Tool Configuration for LABCAR-OPERATOR 3.2.x

	Model ID	Model Data ID	Model Type ID
1	default	default	default
2	EngineModel	default	default
3	IdleControllerHIL	Demo	Demo

Value	
Default Parameter Set	(List)
Project File	IC.lca
Project Folder	C:\Documents and Settings\All Users\Applica
Signal Generator File	StimuliSetInfo.xml
SUT Mapping File	LCOIdleControllerHIL.smf

Default Parameter Set
A list of files containing the default parameters. (*dcm, *.mpa)

Message
Open tcf file C:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.0.0\Examples\Test Bench Configurations\Demo Test Bench\LCO 3.2.x\LCO3.2.tcf

In the left-hand part of the user interface, you can see the “IdleControllerHIL” configuration. A configuration for the LABCAR-OPERATOR tool always consists of three keys: “Model ID”, “Model Data ID” and “Model Type ID”.

The “Model ID” key (with the value “IdleControllerHIL”) references the test project’s UuT “IdleController for Demonstration”. The right-hand part of the window displays the values of this configuration for the “IdleControllerHIL” configuration used in the tutorial project, e.g. the LABCAR-OPERATOR workspace file "ic_520.eew" and the experiment "DefaultExp.eex".

Select **File** → **Exit** to close the editor.

6.4.3 The SUT Mapping File

The SUT Mapping File (SMF) contains a collection of SUT objects. An SUT object describes a mapping of a logical (tool-independent) object to an object in a specific environment. This is how, for example, the logical variable “Engine”, which the Test Case Developer uses in its test cases, is mapped to the “IC/Pedal2NEngine/Limit_EngineSpeed/n_Engine” label of the LABCAR-OPERATOR simulation model.

To open the SUT Mapping File Editor

- Select **All Programs** → **ETAS** → **SMFEditor X.Y** → **SUT Mapping File Editor**.

  The SUT Mapping File Editor opens.

To load the SUT Mapping File

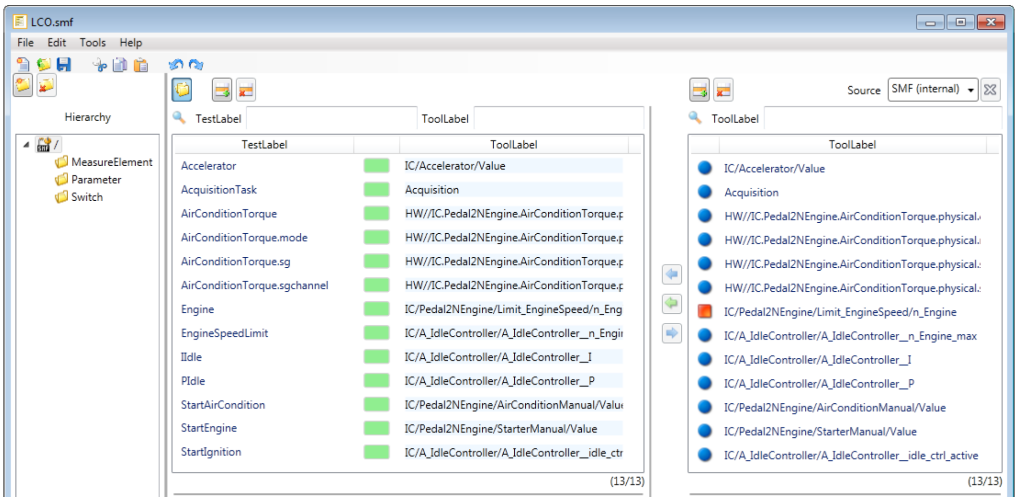
- Select **File** → **Open**.

  A file selector window opens.

- Select the file `<`*Drive*`>:\Users\Public\Documents\ETAS\LABCAR-AUTOMATION 4.2.3\Examples\Test Bench Configurations\Demo Test Bench\LCO 5.x\LCO.smf`.

  The selected SUT Mapping File is loaded.

If you select the top entry in the hierarchy browser, the middle part of the user interface shows all test labels with the relevant tool labels, their type, and hierarchy to which they belong.



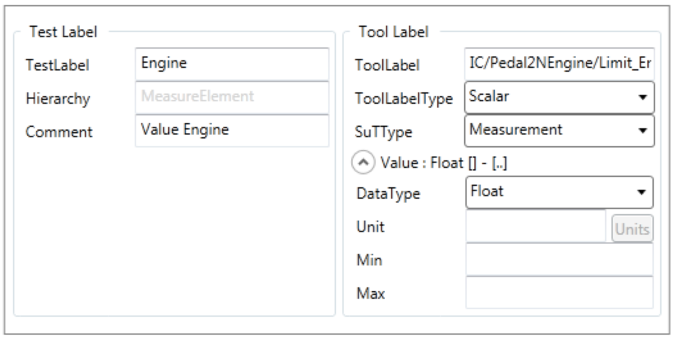
TestLabel		ToolLabel
Accelerator	🟩	IC/Accelerator/Value
AcquisitionTask	🟩	Acquisition
AirConditionTorque	🟩	HW//IC.Pedal2NEngine.AirConditionTorque.p
AirConditionTorque.mode	🟩	HW//IC.Pedal2NEngine.AirConditionTorque.p
AirConditionTorque.sg	🟩	HW//IC.Pedal2NEngine.AirConditionTorque.p
AirConditionTorque.sgchannel	🟩	HW//IC.Pedal2NEngine.AirConditionTorque.p
Engine	🟩	IC/Pedal2NEngine/Limit_EngineSpeed/n_Eng
EngineSpeedLimit	🟩	IC/A_IdleController/A_IdleController__n_Engi
IIdle	🟩	IC/A_IdleController/A_IdleController__I
PIdle	🟩	IC/A_IdleController/A_IdleController__P
StartAirCondition	🟩	IC/Pedal2NEngine/AirConditionManual/Valu
StartEngine	🟩	IC/Pedal2NEngine/StarterManual/Value
StartIgnition	🟩	IC/A_IdleController/A_IdleController__idle_ctr

(13/13)

	ToolLabel
●	IC/Accelerator/Value
●	Acquisition
●	HW//IC.Pedal2NEngine.AirConditionTorque.physical.
●	HW//IC.Pedal2NEngine.AirConditionTorque.physical.
●	HW//IC.Pedal2NEngine.AirConditionTorque.physical.
●	HW//IC.Pedal2NEngine.AirConditionTorque.physical.
🟥	IC/Pedal2NEngine/Limit_EngineSpeed/n_Engine
●	IC/A_IdleController/A_IdleController__n_Engine_max
●	IC/A_IdleController/A_IdleController__I
●	IC/A_IdleController/A_IdleController__P
●	IC/Pedal2NEngine/AirConditionManual/Value
●	IC/Pedal2NEngine/StarterManual/Value
●	IC/A_IdleController/A_IdleController__idle_ctrl_active

(13/13)

If you select an individual SUT object from the table, the properties of the relevant object are displayed in the “Properties” window.



Test Label

TestLabel	Engine
Hierarchy	MeasureElement
Comment	Value Engine

Tool Label

ToolLabel	IC/Pedal2NEngine/Limit_Er
ToolLabelType	Scalar
SuTType	Measurement

Value : Float [] - [..]

DataType	Float
Unit	
Min	
Max	

Select **File** → **Exit** to close the editor.

This completes the description of the three important files for the configuration of the test bench.

7 Glossary

This chapter explains terms which are of particular significance for testing and particularly LABCAR-AUTOMATION 4.2.3.

Abstract Test Case

A → Test Case which is independent of the definition of a concrete → Test Bench.

ASB

→ Automation Sequence Builder

ATCL

The ATCL (Automotive Testing Class Library) is a framework for LABCAR-AUTOMATION for creating test cases that are independent of the relevant test bench. This independence is achieved via → Ports that combine special functionality (→ Signatures) in a suitable way.
For more detailed information, refer to **Programs → ETAS → LABCAR-AUTOMATION 4.2 → Manuals** in "LABCAR-AUTOMATION 4.2.3 - ATCL Getting Started.pdf" and "ATCL Reference Manual.chm".

Automation Sequence Builder

The Automation Sequence Builder is an application for creating test cases.

Automotive Testing Class Library

→ ATCL

Configuration Wizard

The Configuration Wizard is a tool that facilitates the perfect start to developing test cases and creating test projects and test bench configurations.

Control Parameter

Determines how often a → Test Entity is executed.

Control Parameter Specification

XML file which contains sets of → Control Parameters.

Derivate

A derivate is a special parameterization for a UuT within a specific UuT Group. This enables fixed test functionality to be separated from the test parameterization.

ECU

Electronic Control Unit

Global Label List

A collection of labels which can be used in the test process. This list is used by the → Test Case Developer to select the permissible "lbl" attributes for the → LCA Parameters.

Global UuT List

Contains the → UuTs which can be added to a project as well as other information such as "UuT Description" and "UuT Environment Description".

LCA

LABCAR-OPERATOR

LCA Dynamic Array

An (indexed) list of LCA parameters (→ LCA User Records, → LCA Dynamic Arrays and default parameters) of the same type, which can be extended by the → Test Parameter Manager (TPM). Using this type of parameter in an abstract test case makes iterative running of test with different parameter values possible.

LCA Parameter

The parameter type used to parameterize test cases. LCA Parameters contain not only values but also a range of attributes for their description (unit, type, label, comment, etc.).

LCA Test Process Role

Concept and architecture of LABCAR-AUTOMATION 4.2.3 structure the test process into individual activities which are carried out by specific roles (e.g. development of a test case through the role → Test Case Developer). A special tool or interface is made available to each role to carry out the relevant activity.

LCA User Record

A structured parameter type which can be used by the → Test Case Developer to group other parameters (→ LCA User Records, → LCA Dynamic Arrays and default parameters).

Logical Project

A logical project contains all the functionality required to execute a test but no parameterization (apart from standard data set). See → Physical Project.

Master Data Set

The master parameter set of an instance of a → Test Case used in the project. If a new → UuT is added to a → UuT Group, this is initially assigned to the Master Data Set.

Physical Project

A → Logical Project, assigned a (or several) complete parameterization, which is executable.

Port

Abstraction of communication channels for access to a → Test Bench, realized in the form of a collection of function calls (→ Signature).

Project Parameter

Parameters which are available for referencing in the entire test project. This avoids giving ”identical” parameters data on several occasions.

Report Manager

The Report Manager evaluates test results.

Report Viewer

Report Viewer is the application for creating filters for the content and for defining the structure of test reports for the test to be executed.

Signature

Declaration of a function prototype which executes an action on a → Port.

SDF

→ Switch Definition File

SMF

→ SUT Mapping File

SUT

→ System Under Test

SUT Mapping File

An SUT Mapping File (SMF) contains a collection of → SUT Objects.

SUT Object

An SUT object describes a mapping of a logical (tool-independent) object to an object in a specific → Test Bench (e.g. a parameter or a measure value).

Switch Definition File

A file (”lcasdt” extension) which contains a list with definitions of switches. A switch is a parameter of the type ”enumeration” which contains a set of descriptive values. The → Test Case Developer uses the switches defined there, the → Test Parameter Manager (TPM) sees the possible positions of a switch and the → Test Bench Configuration Manager (TBCM) knows which mappings it has to provide for it in the → SUT Mapping File.

System Under Test

The entire test structure (→ UuT and → Test Bench).

Test

... a process of planning, preparation, and measurement aimed at establishing the characteristics of an information system and demonstrating the difference between the actual and the required status (taken from: T. Koomen and M. Pol, Test Process Improvement: A Practical Step-by-Step Guide to Structured Testing, Addison-Wesley (1999), ISBN 0-201-59624-5)

Test Architecture Description

XML file which contains the → Ports and → Signatures used in the → Test Case.

Test Bench

The entire environment in which the test is executed consisting of hardware, software, test tools, procedures etc.

Test Bench Configuration Manager (TBCM)

The task of the Test Bench Configuration Manager is the provision and configuration of all tools to make a certain test project executable.

Test Bench Connector for LABCAR

Test Bench Connector for LABCAR is an application for connecting the experiment environment LABCAR-OPERATOR.

Test Bench Initialization (TB Init)

The process of the transition of the Test Bench tools to the state "Tool Configured" (before the test sequence is carried out). All "ConfigureTool" signatures of the ports used are executed on the Test Bench tools using the "UuT Environment Description" of the selected UuT.

Test Sequence

A specified execution sequence of physical $\rightarrow$ Test Cases.

Test Sequence Manager (TSM)

After the Test Parameter Manager has created the physical test project, it is the task of the Test Sequence Manager (TSM) to specify execution orders for test runs.

Test Case

The basic element of the $\rightarrow$ Test Release Library, consisting of hierarchy descriptions ($\rightarrow$ Test Hierarchy Description), parameters ($\rightarrow$ Test Parameter, $\rightarrow$ Control Parameter) and $\rightarrow$ Ports and $\rightarrow$ Signatures used ($\rightarrow$ Test Architecture Description) and possibly an executable.

Test Case Developer

The Test Case Developer is not part of the task list of LABCAR-AUTOMATION 4.2.3. It develops Test-Bench-independent (abstract) and logical $\rightarrow$ Test Cases and provides these in a $\rightarrow$ Test Release Library for use in test projects.

Test Design Tool

Development tool for creating $\rightarrow$ Test Cases.

Test Entity

A hierarchy node in the functional hierarchy of a test case.

Test Handler

Test Handler is the application in which

– the $\rightarrow$ Test Handler (TH) determines the procedure of tests defined for a specific $\rightarrow$ UuT within a test project on a specific $\rightarrow$ Test Bench

and

– the $\rightarrow$ Test Bench Configuration Manager (TBCM) creates and manages the Test Bench Configurations which, if necessary, can be selected by the $\rightarrow$ Test Handler (TH).

Test Handler (TH)

Once functionality, parameterization, order of execution and test environment are defined, the Test Handler (TH) executes the test on a specific $\rightarrow$ Test Bench.

Test Hierarchy Description

XML file which contains a description of the hierarchical structure and the functionality of the $\rightarrow$ Test Entity.

Test Manager

Test Manager is the application for creating and managing test projects, parameterizing tests and creating $\rightarrow$ Test Sequences.

Test Parameter

→ LCA Parameter

Test Parameter Manager (TPM)

Once the Test Project Manager has created the logical test project, it is the task of the Test Parameter Manager (TPM) to transform it into a → Physical Project.

Test Parameter Specification

XML file which contains sets of → Test Parameters.

Test Project

A test project contains all information which is necessary for executing a test. This information is specified by the different roles.

Test Project Manager (TPrM)

The Test Project Manager (TPrM) creates → Test Projects, manages the → UuTs and the test functionality which are assigned to these.

Test Release Library

In this library, released, logical → Test Cases, including the XML files which describe the hierarchy (→ Test Hierarchy Description), parameters (→ Test Parameter, → Control Parameter) and the → Ports and → Signatures used (→ Test Architecture Description) of each → Test Case.

Test Specification

User Requirements Document for test – this results in the → Abstract Test Case.

Testing and Test Control Notation

Testing and Test Control Notation (TTCN-3) is a language for specifying and implementing tests.

Tool Adapter

A DLL which realizes access to a special tool (e.g. LABCAR-OPERATOR) for an individual → Port by calling → Signatures.

TTCN

→ Testing and Test Control Notation

TRL

→ Test Release Library

Unit Conversion Table

A file which contains a list of units and conversion formulae for these to be able to convert values with certain units as they are used by the Test Bench.

Unit under Test

The object to be tested, the test specimen – typically an ECU.

UuT

→ Unit under Test

UuT Group

A → Test Project consists of one or more UuT Groups which in turn consist of the → UuT itself, → Test Cases, → Test Parameters and → Test Sequences.

UuT Structure

A structure for the hierarchical grouping and structuring of a set of UuTs in a test project. The UuT Structure can be created and managed by the → Test Project Manager (TPrM).

Value Constant

A constant definition valid for the entire project which can be created by the → Test Parameter Manager (TPM). Value Constants can be assigned to scalar parameters (like in C) so that no longer the value but just the name of the constant is displayed in the user interface.

8 ETAS Contact Addresses

ETAS HQ

ETAS GmbH

Borsigstraße 14	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	WWW:	www.etas.com

ETAS Subsidiaries and Technical Support

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:

ETAS subsidiaries	WWW:	www.etas.com/en/contact.php
ETAS technical support	WWW:	www.etas.com/en/hotlines.php

Figures

Index