
LABCAR-OPERATOR V5.4.0

Release Notes

Copyright

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

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

© **Copyright 2015** ETAS GmbH, Stuttgart

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

Release Date: December, 15th 2015

Thank you for using our products. This document contains important information. It updates the documentation of LABCAR-OPERATOR V5.4.0 and the corresponding add-on products, reflecting known important software and documentation issues. We strongly recommend that you read the entire document.

We welcome your comments and suggestions.

Stuttgart, December 8th, 2015

Contents

1	Introduction	6
2	Product Definition	6
2.1	Functions at a Glance	6
2.2	General Description	6
2.2.1	System Prerequisites	6
2.2.2	Software Prerequisites	7
2.3	Delivery	8
3	Installation	9
3.1	Additional Installation Hints	9
3.2	Licensing	12
3.2.1	Software Disclaimer for ASAM AE HiL and XiL APIs	12
4	Changes	13
4.1	New Features	13
4.2	Known Issues	13
4.2.1	Known Issue Reports	13
4.2.2	Software related items	13
4.2.3	Hardware related items	37
4.3	Compatibility issues to projects created with earlier versions of LABCAR-OPERATOR	38
4.4	Discontinued Features	41
5	Hints	43
5.1	User Rights to use LABCAR-OPERATOR	43
5.2	Issues with LABCAR-CCI (Calibration Connector for INCA)	44
5.3	Issues operating LABCAR-OPERATOR and INCA in parallel	45
5.4	Error message "libut.dll not found" after installation of MATLAB®/Simulink®	46
5.5	Multiple Versions of The MathWorks Releases on one PC	46
5.6	Use of LABCAR- In-/Outport Blocks for Simulink® models without having LABCAR-OPERATOR involved	47
5.7	How to remove missing Ethernet board from WINDOWS Device Manager	47
5.8	RTPC simulation target: Rules for OS setting in LCO	47
5.9	Problems when assembling Ethernet components with different transmission rates: Operation "Save parameters" doesn't work	48
5.10	Different subscription of array indices in LABCAR-OPERATOR Experiment Environment and LABCAR-PARAMETRIZATION ASSISTANT	48
5.11	Issue with borrowing of LCO concurrent license	48
5.12	Deactivation of PC Power Management for unattended operation	48
5.13	Open a project without a license: IP is closed without information	49
5.14	FlexRay™ Residual Bus simulation with EB tresos® Busmirror® containing instances of signals	49
5.15	Change of FlexRay™ Tool Chain Version triggered only for supported versions	51
5.16	Profiling of Simulink® Subsystems (EHI # 314964)	51
5.17	Avoid Tesis model license error	54
5.18	Limits for curves and maps in Experiment Environment	54
5.19	Issues with LABCAR-LCE/LCX (LABCAR Integration ETK/XETK)	54
5.20	Issue on play back of scripts with Experiment Environment Script Recorder	54
5.21	"Global Parameters" in LABCAR-MCS based LCO projects (Globally shared workspace names)	55
5.22	How to update link for Simulink Module imported as reference	56
5.23	How to change reference behavior of Simulink Model imported as link	56
5.24	Meaning of checkbox settings for code generation	56
5.25	How to exclude files to be copied to codegen directory and RTPC	58

6	Glossary and References.....	59
6.1	Definitions and Abbreviations	59
7	ETAS Contact Addresses	61

1 Introduction

LABCAR-OPERATOR is a modular experiment environment for ECU testing. This document describes the properties of LABCAR-OPERATOR V5.4.0, which replaces all previous versions.

This release notes document contains:

- General product information
- Known issues
- Hints

2 Product Definition

2.1 Functions at a Glance

- Efficient and easy operation through user-oriented separation into integration platform and experiment environment
- Experiment environment with a modern, unified, and ergonomically designed operating concept
 - Easy navigation through unified element tree with convenient and adjustable search options
 - Display layers for rapid switching between different experiment views
 - Optimized adaptation to individual use cases through numerous setup options, e.g., "User-defined labels" and instrument properties
 - Newly designed, flexibly adaptable automotive instruments and library
 - Large variety of analysis options through professional oscilloscope with negligible processor load
- Easy integration of the most diverse environment models based on MATLAB®/Simulink® ASCET or C code. To operate LABCAR-OPERATOR experiments, the design tools—and applicable licenses— are not required.
- Enhanced CAN bus simulation with message and signal manipulation, plus CAN monitoring
- HiL system is quickly adapted to different ECU variants
- Integration of measurement and calibration tool INCA by ETAS

2.2 General Description

2.2.1 System Prerequisites

The following system prerequisites have to be met:

	Minimum Requirements	Recommended Equipment
Hardware	2 GHz Pentium Dual Core PC 4 GB RAM Ethernet Adapter 10/100BaseT DVD-ROM drive Graphic Adapter with DirectX 9 or higher Resolution 1280x1024, 16 bit color Hard disc min 20 GB	2 GHz Pentium Quad Core PC 8 GB RAM 2 nd Ethernet Adapter 10/100BaseT DVD-ROM drive Graphic Adapter with DirectX 9 or higher and hardware acceleration Resolution 1600x1200, 512 MB, 32 bit color
Operating System	Windows® 7 32bit SP1	Windows® 7 64bit SP1
Free Disk Space	>5 GB	>5 GB

2.2.2 Software Prerequisites

LABCAR-OPERATOR Compatibility List

For parallel operation, the tool versions as given in the LABCAR-OPERATOR Compatibility List are recommended. The list shows that combination of tools, which have passed through our predefined release procedure.

You may find the LABCAR-OPERATOR Compatibility List related to the current version(s) in the product documentation on the product media (file "LABCAR-OPERATOR V5 - Software Compatibility List yz.pdf") as well as via the user manuals of the installed software.

Updates of the LABCAR-OPERATOR Compatibility List that might occur either due to releases of new versions of a supported tool between two releases of LABCAR-OPERATOR software or by change of support in LCO due to bug fix will be announced in a new versions of "LABCAR-OPERATOR V5 - Software Compatibility List.pdf" and distributed together with regular hotfix packages issued for LABCAR-OPERATOR. An ordered number indicating the update of the compatibility list through all LCO V5 versions will be included within the pdf itself.

IMPORTANT:

LABCAR-OPERATOR V5.4.0 is delivered with some base components of INCA V7.1.10. This components can cause malfunction of INCA 7.1.9 and older. It is highly recommended to upgrade INCA V7.1 to INCA V7.1.10 (V7.1 SP10) at least even if LABCAR-OPERATOR V5.4.0 is not used in parallel with INCA V7.1.

LABCAR-OPERATOR V5.4.0 add-ons and required tools

The different Add-Ons of LABCAR-OPERATOR require additional software. The version supported is indicated in the "LABCAR-OPERATOR Compatibility List" (see section above):

LABCAR-OPERATOR general	Microsoft® .NET Framework V4.5.0 (includes V2.0) Redistributable Package (included on the DVD)
LABCAR-MCS (Modeling Connector for Simulink®)	One of the supported releases of MATLAB®/SIMULINK® by The MathWorks. Requires Matlab and Simulink Coder. 32 bit and 64 bit versions are supported.
LABCAR-CCI (Calibration Connector for INCA®)	One of the supported INCA® releases by ETAS.
LABCAR-MCA (Modeling Connector for ASCET®)	One of the supported ASCET® MD releases by ETAS.
LABCAR-NIF (Network Integration FlexRay™)	EB tresos® Busmirror , configuration tool for FlexRay™ residual bus simulation by Elektrobit.
LABCAR-NIL (Network Integration LIN)	LIN specification V1.3, V2.0, V2.1 and V2.2 is supported. No external tool required.
LABCAR-NIC (Network Integration CAN)	CAN specification as interpreted by CANdblib V3.2 by Vector Informatik. No external tool required.
LABCAR-LCX (LABCAR)	The A2L file imported as configuration file for the module need to provide values for

Integration XCP)	<p>communication parameters for the XCP connection (section "IF_DATA_XML" in A2L file) and a section describing the interface structure AML for XCP V1.0.1" or V1.1.0). It should support A2L standard version V1.6.</p> <p>For stimulation of measurements hooks need to be implemented to enable hook based bypass with X-ETK T2.1.</p>
LABCAR-LCE (LABCAR Integration ETK)	<p>For ETK connection, the A2L files have to comply with AML V1.2 - V1.7 and ASAP2 V>=1.40. Additional hardware ES592, ES910.2 or ES910.3 for connection ETK with RTPC-PC is required (ES910.1 cannot be used).</p> <p>For stimulation of measurements hooks need to be implemented to enable hook based bypass. For stimulation of scalar parameters a parallel ETK is required.</p>

Our in-house release tests have been passed with Tesis® Dynaware® models enDyna 2.0.7 Release 3.3.8 (for more information about compatibility of the most recent releases please refer to the Tesis® Dynaware® documentation) and ETAS Simulation models (discontinued) GEVM V5.0.2/DEVM V5.0.1.

2.3 Delivery

The software is delivered with an installation routine on a DVD including LABCAR software, documentation, tools, utilities, and further information. All software documentation is available in the Portable Document Format (PDF), which requires Adobe® Reader®. You can download the actual version from Adobe® (<http://www.adobe.de/products/acrobat/readstep2.html>).

The DVD contains the following items in the folder 'data':

Directory	Meaning / Explanation
HSP_Installation	Hardware Service Pack
LABCAR-ASC	ASCET AddOn for LABCAR-OPERATOR
LCO_Installation	LABCAR-OPERATOR V5.4.0 setup
Licensing	Applications and information regarding ETAS licensing
Misc	Contains examples and additional files providing help to the user. See the file "LABCAR-OPERATOR V5.4.0_Examples_Readme.pdf" in the folder for details about the contents.

3 Installation

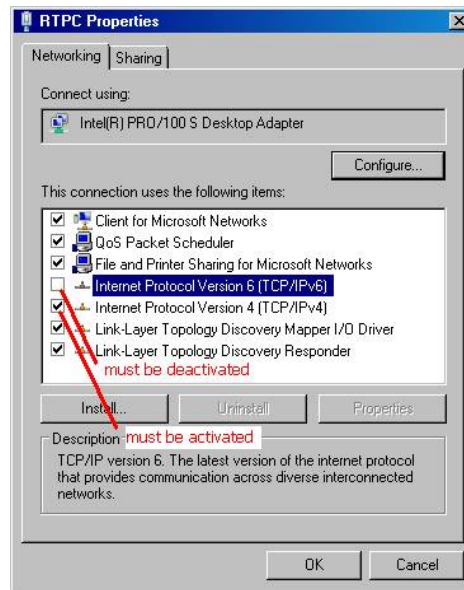
To install LABCAR-OPERATOR please follow the guidelines described in chapter 3 "Installation" of the manual "**LABCAR-OPERATOR V5.4.0 – Getting Started/Schnelleinstieg**" provided within the Manuals -> Software folder of the installation DVD. Beside the information about installation of LABCAR-OPERATOR you will find information on firmware update for LABCAR hardware and on flashing the ES1130 VME bus controller.

3.1 Additional Installation Hints

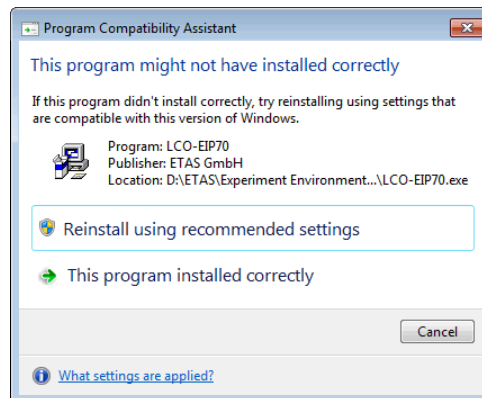
- Do not mix products and/or add-ons of different versions of LABCAR-OPERATOR software. However it is possible to install LABCAR-OPERATOR V5.1.x, V5.2.x, V5.3.x and LABCAR-OPERATOR V5.4.0 on the same PC.
To make this possible we provide an API Switching Tool (via Start Menu Programs/ETAS/LABCAR-OPERATOR 5.4/LABCAR-OPERATOR 5.4 Tools => LABCAR-OPERATOR Version Selector) in which you can select the version you want to work with. It is not possible to install multiple LCO V5.4.x versions in parallel.
Before start installation, shutdown all running applications (especially all ETAS applications have to be closed).

Hint: Since older LABCAR-OPERATOR versions do not support the minimum required Windows™ 7 SP1 they are not listed here anymore.

- For silent installation, there is a documentation file setup.pdf in the "~Data/Misc/Silent Installation" subfolder on the product DVD. In this file, several typical silent installation tasks are listed.
- For installation of LCO software administrator privileges are required. For operating of LCO special user and access rights need to be granted. See chapter 5.1 "User Rights to use LABCAR-OPERATOR" for more details.
- Before installation of LABCAR-OPERATOR the installation of additional tools (like Microsoft .NET Framework, Microsoft Visual C++ Runtime Library and others) is required. The installation takes care of this. One may find the installation files of these prerequisites on the LCO installation DVD.
For Korean installation only: After installation of Microsoft .NET Framework a reboot of the PC is required.
- Make sure that the correct HSP version is installed on board ES1130, use HSP provided with this LABCAR-OPERATOR software package to transfer an empty project in the ES1130 program ROM.
- Known issues with Windows™ 7:
 1. Prefer installation of LABCAR-OPERATOR with administrator rights. If you start the file „setup.exe“ without administrator rights the installation process is not finished completely successful. If you are logged in without administrator rights perform the installation by executing the file „setup.exe“ from the WINDOWS Explorer's context sensitive menu "Run As..." and an account with administrator rights. Do not start the installation by means of the HTML Page Setup what is started automatically when you insert the DVD in the PC.
 2. Make sure that only the Internet Protocol Version V4 is activated for the Ethernet adapter which is used to connect to the RTPC PC (LCO-IP and LCO-EE is concerned). The Wake On LAN feature does support IPv4 only.



3. During installation one or more confirmation windows may be displayed similar to the following:



These windows may be displayed during installation in case nothing is changed in the Windows System Registry when installing the prerequisites but they do not block installation progress even on silent installation. Just confirm the window with "This program installed correctly" or "OK" to finalize the installation.

- **DirectX Installation Fails (EHI # 99822)**
A pre-requisit for installing and running EE 3.7.0 is to install DirectX Runtime Components. This is installed automatically when the EE 3.7.0 is installed. However it is possible that the installation will fail on some computers. If the installation does fail it could be that the following registry entry has an incorrect value

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust
Providers\Software Publishing]

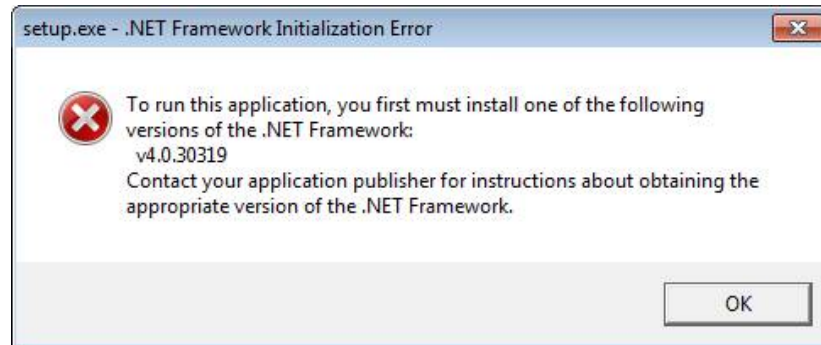
The correct value should be "State"=dword:00023c00.

In addition the installation fails if DirectX11 or higher is already in place. In this case you need to trigger the DirectX9 installation manually. The installation can be found on the DVD in the following folder: \Data\LCO_Installation\Packages\3rdParty DirectX9_1.0.0.1\DXSetup.exe.

- After successful installation it is recommended to assign the correct INCA® and Simulink® tool versions by means of the provided tools "Associate to INCA®" in the menu START => All

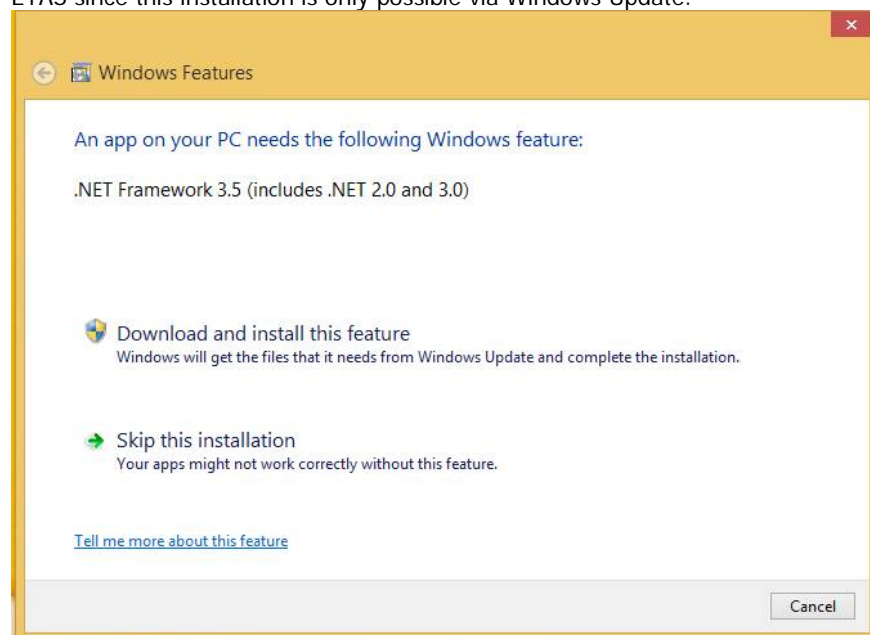
Programs => ETAS => Experiment Environment 3.7 => Experiment Environment 3.7 Tools, "Matlab Version Selector" in the menu START => All Programs => ETAS => LABCAR-OPERATOR 5.4 => LABCAR-OPERATOR 5.4 Tools.

- Known Issue with the LABCAR-OPERATOR V5.4.0 installer
If you execute the "Setup.exe" of the LABCAR-OPERATOR V5.4.0 you may get the following message box.



Since the installer needs .NET Framework 4.0 to be executed you need to install it first. The installer is shipped with the Product DVD and can be found in the folder "~Data/Misc/.NET 4.0 Installation".

- Known Issue with Windows™ 8.1:
During the installation you may be asked to install .NET Framework 3.5. This cannot be avoided by ETAS since this installation is only possible via Windows Update.



If this dialog box is shown you need an active internet connection and click "Download and install this feature" before the installation can be completed and the software can be used.

3.2 Licensing

Please note, that LABCAR-OPERATOR V5.4.0 requires software license activation. Besides the perpetual product licenses that you obtain, when you purchase LABCAR-OPERATOR, ETAS offers fully functional time-limited evaluation licenses.

In order to obtain your software license certificates, please start the "LicenseInfo" application from the Product DVD (Data/Licensing/LicenseInfo.exe). It will provide you with a list of network adapters installed in your system. Please choose an adapter from the list. It will later be referenced by your license certificates to ensure that they are valid for the respective machine. We recommend you to choose an adapter that is always present in your system (e.g. the main company network adapter). Please make sure, that this adapter is also present, if you remove e.g. your laptop from the docking station.

After selecting the network adapter, the tool will provide you with information on the used MAC address and the user name.

During the ordering process, ETAS provides you with a license activation number (the so-called Entitlement ID). With this information, the MAC address and the user name, please do one of the following:

- Visit <http://www.etas.com/support/licensing> and generate your license certificate based on the information mentioned above.
- Mail the information mentioned above to licenses.de@etas.com or one of the contact addresses provided in Section 7.

The information that you submit permits ETAS to generate the software license certificate, i.e. the license key. It will not be used for any other purpose.

Please copy the license key into a text file with the extension *.lic (e.g. LABCAR.lic) and store it on your hard disk. When starting LABCAR-OPERATOR, it will ask you for the location of this file. If you have further questions regarding the installation procedure, please refer to the "Getting Started" manual or contact ETAS for assistance.

3.2.1 Software Disclaimer for ASAM AE HiL and XiL APIs

The LCO product medium provides the installation for ASAM AE HiL API V1.0.2 and XiL API V2.0.1 in the folder "\\Data\Misc\ASAM HiL and XiL APIs". ETAS draw your attention to the fact that installing LABCAR-OPERATOR software also installs components that are licensed and released by ASAM e.V. (Association for Standardization of Automation and Measuring Systems). ETAS here by confirms to be a legal licensee of ASAM and as such entitled to use these licenses and to install the ASAM software together with the LABCAR-OPERATOR software. The customer is not authorized to pass the ASAM binaries to third parties without permission.

For further information, see <http://www.asam.net>.

4 Changes

Starting with LABCAR-OPERATOR V4.0 there are two separate applications in LABCAR-OPERATOR software: For HiL system configuration, the Integration Platform (IP), and for executing tests and experiments, the Experiment Environment (EE). While the IP is based on the previous LABCAR-OPERATOR V3 program window, the EE is a completely new development based on the INTECRIO EE.

4.1 New Features

For detailed information about features introduced in LABCAR-OPERATOR V5.4.0 please refer to the document "LABCAR-OPERATOR V5.4.0 – What is new.pdf".

4.2 Known Issues

This section describes known restrictions of the released version LABCAR-OPERATOR V5.4.0.

4.2.1 Known Issue Reports

If a product issue develops, ETAS will prepare a Known Issue Report (KIR) and post it on the internet. The report includes information regarding the technical impact and status of the solution. Therefore you must check the KIR applicable to this ETAS product version and follow the relevant instructions prior to operation of the product.

The Known Issue Report (KIR) can be found here: <http://www.etas.com/kir>

4.2.2 Software related items

Issues using LABCAR-OPERATOR on Windows 8.1

Repeated automated tests may cause application crashes in LABCAR-IP and LABCAR-EE

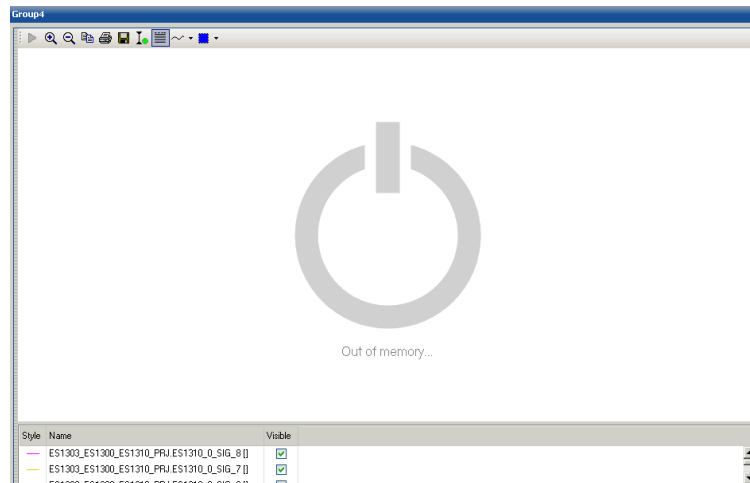
During automated testing, which repeatedly open and close both LABCAR-IP and LABCAR-EE, in rare cases application exceptions in LABCAR-IP and LABCAR-EE have been observed. These exceptions can lead to application crashes. The situation can be recovered by closing and re-opening the applications again.

We are working on a solution for these issues and will ship updates to solve them in the next months.

Platform Oscilloscope Issues

Insufficient Graphics Card Memory

The Platform Oscilloscopes directly accesses the graphics adapter via DirectX. Therefore, it needs graphic card memory (Video RAM). The size of the required Video RAM depends on the number of Platform Oscilloscopes which are visible and their size. If the Video RAM is not sufficient, the Platform Oscilloscope will display a message similar to the one shown below:



In that case, you might free Video RAM by closing other running software, or by decreasing either size or amount of Platform Oscilloscopes shown at the same time, e.g. by moving some to another layer. Once there is sufficient memory again available, the Platform Oscilloscopes will turn into normal operation mode automatically.

Problems with Screensaver during endurance tests

During endurance tests some interference problems between the Windows screensaver and the Platform Oscilloscope leading to a freeze of the Platform Oscilloscope have been observed. Therefore, it is advised to disable the screen saver when performing endurance tests.

Problems with time consumption actions within Experiment Environment

During actions like "Stop Datalogger" (=> saving results to harddisk) or "Reconfiguration of Signal Generator" (=> recreation of data elements on RTPC target) freezes of the Platform Oscilloscope have been observed. We currently working on a successor of the Oscilloscope.

Signal Generator

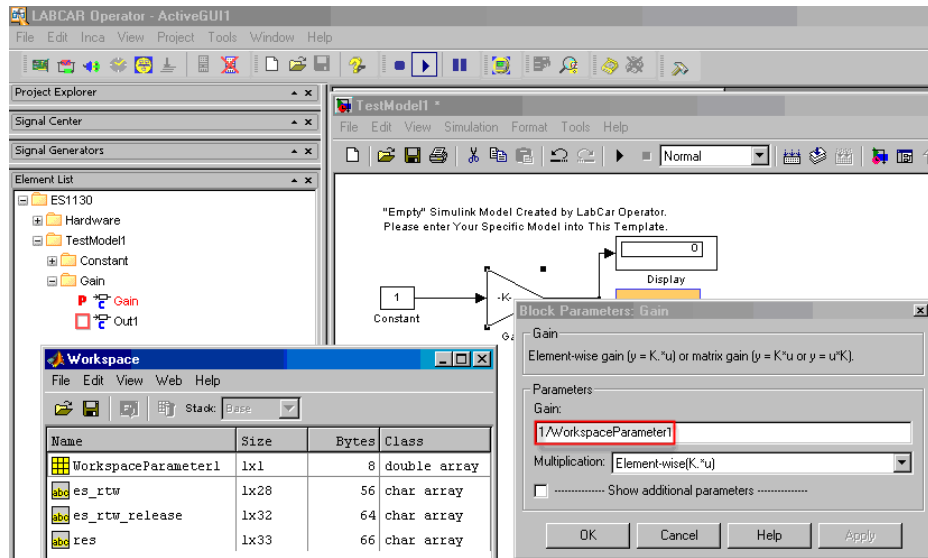
Exception in Signal Editor preview (EHI # 253664)

The Signal Description Preview Signal Editor uses DirectX technology. We encountered initialization and rendering errors caused by old graphics device drivers, especially Intel Express Chipsets. This may result in an Exception Dialog showing up or a corrupted drawing of the Signal preview (a red X on a white background). An update of the graphics adapter driver solves the issue. If this problem occurs, please update the graphics card driver to the newest version.

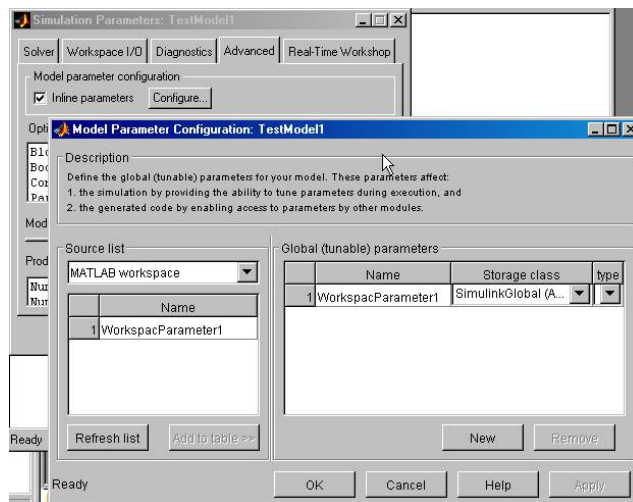
Simulink® related issues

Algebraic expressions in the input field of mask parameters of blocks in the Simulink® model

Sometimes algebraic expressions are used in the mask parameter input field of blocks in a Simulink® model. If these algebraic expressions contain a global workspace parameter, then the handling of these algebraic expressions during code generation depends on some additional settings. Have a look at the picture below:



The (global) workspace parameter “WorkspaceParameter1” is referred to in the algebraic expression “1/WorkspaceParameter1”. The evaluation of the expression on code generation depends on the setting “Inline Parameters” (picture shows example for an older Simulink version):



If the checkbox “Inline parameters” is inactive, the formula “1/WorkspaceParameter1” is pooled and the result of the division is assigned as initial value to the mask parameter “Gain”. This (result) parameter “Gain” is shown in the LCO-EE Workspace Elements tree afterwards and the information about the algebraic expression containing “WorkspaceParameter1” is lost.

If “Inline Parameters” is set to “Yes” and the independent parameter “WorkspaceParameter1” is added to the list “Global (tunable) parameters” (as shown in the picture above) the algebraic expression is integrated in the code as formula. In this case again after code generation the parameter with the mask name “Gain” is displayed in the LCO-EE Workspace Elements tree. But now it refers to the storage class originally assigned in the model to the global workspace parameter “WorkspaceParameter1”. This leads to different results for the output of the GAIN block. In the first case the output is “Gain” * input in the latter case the output is (1/GAIN) * input. We recommend using S-Functions instead of Gain blocks, in case masking and workspace variables are used.

Names of masked / workspace parameters defined as “Global Parameters”

In MATLAB®/Simulink® there are two ways to utilize parameterization for LABCAR-OPERATOR. Parameters can either be declared as “masked” or the mechanism of workspace variables can be used.

The name of a parameter, as shown in the LABCAR-EE Workspace Elements tree, is generated as follows:

In case a parameter was set to “masked”, the mask name is used; in other cases the workspace variable name is taken.

Using pre-defined blocks from Simulink® (e.g. a “Constant” Block) results in masked parameters by default and consequently the masked name e.g. “Value” or “Gain” is used. If this parameter occurs at least one more time in the model, it is assigned to the folder “_Global Parameters”. If the workspace variable name for some reason needs to be used, insert an unmasked block (e.g. an S-Function) to avoid the automatic name generation of masked blocks. (EHI # 317079)

Problems with library containing Simulink™-RTW code created with a previous LCO/RTPC version (error message “Compile: rt_atan2.c in file included from rt_atan2.c:15: rtplibsrc.h:18: rtwtypes.h: No such file or directory”, EHI # 314688)

When building a LABCAR-OPERATOR project with a new LABCAR-OPERATOR and/or a new RTPC version an error message similar the following may displayed in the LCO build log and the build cannot be finished successfully:

Build log

```

Timestamp: 2006-11-07, 10:32:20
Project: TOP_55
Target: RTPC
Codegen path: c:/lc_xt_projects/target_rtpc/codegen/
Raw log file: c:/lc_xt_projects/target_rtpc/codegen/build.log
Compiler warnings: 3
Errors: 2
-----
Enter: .
RTPC: Telnet connection to 192.168.40.14 established
labcar@rtpc:~/simulink_files
Compile: rt_atan2.c
In file included from rt_atan2.c:15:
rttplibsrc.h:18: rtwtypes.h: No such file or directory
RTPC: Created Matlab library.
RTPC: Upload.
Enter: .
Enter: .
Enter: ./asap
Enter: ./asap/src

```

The reason for that may be that the library (tar file) which includes C and H files delivered by Simulink® was created with an LCO version before V3.2.0. At that time the TAR file missed the file "rtwtypes.h" if no license of Stateflow® was available. This resulted in the compiler error message as shown in the picture. To enforce the re-generation of the library erase the following files:

- a) In the LCO project directory “[LCO project name]\Target_[target kind]\SimulinkModels\libSrc” the file “[version].tar.gz” (for example “r14sp3.tar.gz”).
- b) On the RTPC: All files in folder “home/labcar/lib” and in folder “home/labcar/simulink”. This need to be done for target RTPC only.

Initialization problems with S-Function inputs of ETAS simulation models when using MATLAB®/SIMULINK® version R2007a or higher (EHI # 321910)

Interference when setting inputs of C code S functions to constant values connected to these inputs with the code programmed for the initialization routine in the s function was observed in the following conditions:

- 1. Use of MATLAB®/SIMULINK® version R2007a or higher

2. A constant Block is connected to an input of a S-Function created with S-Parser
3. The optimization option 'inline parameters' has been switch on

When starting the simulation, the code defined for the function 'mdlInitializeConditions' overwrites the constant values prior assigned by the Simulink-Engine.

We recommend replacing the Constant blocks by LABCAR-Inports to ensure correct initialization of the S function inputs. You may also configure a sample time for the Constant blocks to reassign the initialization order.

Black list of MATLAB®/Simulink™ blocks currently not supported by LABCAR-OPERATOR

Blocks with Absolute Time

The Real-Time Workshop cannot process blocks which use absolute time (e.g. "Sine Wave", "Clock", "Pulse Generator", "Signal Generator", "Transport Delay").

Blocks for which no Parameters and Outputs are Generated

No parameters or outputs are created for a range of model blocks - this applies to the following blocks:

- Scope
- Floating Scope
- Display
- XY Graph
- Inport
- Outport
- Memory
- Goto
- ToWorkspace
- Stop
- FromIf
- Ground
- EnablePort
- GotoTagVisibility
- TriggerPort
- SignalSpecification
- Selector
- Mux/Demux
- BusCreator/BusSelector
- Terminator

Blocks which access the MATLAB® interpreter

To avoid timing conflicts there is no MATLAB® Runtime Engine on the RTPC during execution time. Therefore M script instructions invoking the MATLAB® interpreter are not supported. This applies to the following block:

- M file S functions

Blocks which turned out to cause problems during build

Considering the increasing number of blocks provided by MATLAB®/SIMULINK and AddOn Blocksets it is not feasible to test all of them for each release in advance. In the following list you can find blocks where build problems have been reported with earlier releases of MATLAB®/SIMULINK and/or LABCAR-OPERATOR. We will investigate all upcoming problems to remove blocks from this list as soon as possible:

- Model Reference Block
- ForEachIterator

Parallel installation of 32-bit and 64-bit of the same MATLAB®/SIMULINK™ version

Currently the 32-bit and 64-bit version of the same MATLAB®/SIMULINK® (e.g. R2014a) share the same root registry identifier. Therefore it is not possible to distinguish which version should be used. In such a case Windows itself decides which version is called. The decision is made by the architecture of the calling program (x86 or x64). Since LABCAR-OPERATOR V5.4.0 is still based on the 32-bit architecture (x86), Windows will force us to use the 32-bit version of MATLAB®/SIMULINK®.

Therefore you need to uninstall the 32-bit version of your MATLAB®/SIMULINK® if you want to make sure that the 64-bit version is used. In addition to this restriction we will not offer the 64-bit version of MATLAB®/SIMULINK® within the "Matlab Version Selector" as long you have the corresponding 32-bit version installed.

No 64-bit compiler is delivered with MATLAB®/SIMULINK™ (64-bit)

With MATLAB®/SIMULINK™ no 64-bit compiler is included which is needed to use the Real-Time Workshop or to access mex64 files within your model. Please find all necessary steps to integrate a 64-bit compiler on your system at <http://www.mathworks.com>.

During build 'rtCP_pooled_' errors linked to the const_params.c file of MATLAB® occur

This error appears only if:

1. at least two SIMULINK™ models are integrated into a LABCAR project and
2. at least two SIMULINK™ models have to have the "Simulation -> Configuration Parameters -> Optimization -> Signals and Parameters -> Inline Parameters " checkbox checked and
3. both SIMULINK™ models have SIMULINK™ blocks (constant, gain, etc...) with numeric values or non-inlined workspace variables inserted

To solve this problem you could do the following:

1. If the SIMULINK™ models can be merged, create only one model if possible.
2. Uncheck "Simulation -> Configuration Parameters -> Optimization -> Signals and Parameters -> Inline Parameters " for the models where this is not needed. This, however, will deactivate the listing of workspace parameters, which are handed over to several S-functions in one model as "_GlobalParameter" within LABCAR-EE. Hence, if the inlining of parameters is needed for more than two models you have the following options:
 - a. For MATLAB® R2012b and R2013a:
Follow the bugfix of MathWorks® at:
<http://www.mathworks.com/support/bugreports/919341>
Note: Be aware that you need an MathWorks® user account to access this bug report.
 - b. For MATLAB® R2013b and R2014a:
All the SIMULINK™ blocks (constant, gain, etc...) within your models with numeric values or non-inlined workspace variables must get values through workspace variables that are set to globally tunable.
To do this, follow the following steps:

- Identify all the blocks in the models

- Pass instead of the value a workspace variable name (that you previously declared)
 - Go to “Simulation -> Configuration Parameters -> Optimization -> Signals and Parameters -> Inline Parameters”
 - Identify in the source list column the workspace variable that you declared and passed as value to the constant blocks
 - Move the variable to the global (tunable) parameters
 - Load the new created workspace variable (myzero, myhundred, etc) (via i.e. ModelPreload.m) into the workspace
 - Delete slprj and %ModelName%_LABCAR_rtw within LCO project directory (%LCO_Project%\Target_%Name%\SimulinkModels\%ModelName%)
 - Generate code with SIMULINK™ RTW
 - Inspect if the file const_params.c is still present at %LCO_Project%\Target_%Name%\SimulinkModels\%ModelName%\slprj\lcr\sharedutils\
 - If yes, you forgot some replacements => please start over again
- c. For MATLAB® R2014b and R2015a:
Modify the PreLoad Callback of the Simulink Model and add the following MATLAB® code:

```
if ~verLessThan('Simulink','8.4')
set_param(bdroot, 'UtilityFuncGeneration', 'Shared location')
set_param(bdroot, 'GenerateSharedConstants', 'off')
end
```

Not all old Simulink™ models with Dynamic Linked Library S-Functions can be built successfully anymore

Some SIMULINK™ models uses Dynamic Linked Library S-Functions. These are basically always linked to specific Microsoft C++ libraries. Not all different C++ libraries are always present on any system. Therefore we shipped a set of different Microsoft C++ libraries via the Product DVD. These libraries including a readme can be found within “~\Data\Misc\MATLAB Simulink - Dynamic Link Library S functions”.

ASCET® related issues

MPA parameter file format does not support ASCET Parameter “characteristic curves/maps” where the data types for input/output quantities are different

With the current version of MPA file format the ASCET parameter of object type 1D- and 2D-TABLE are not supported in the special case that the x/y/z (axes of supporting points and data type of curve/map values) are not of the same data type. We recommend to save parameters of this kind in DCM format.

Build problem due to missing header file

If in ASCET the External Source Editor for C-Code-Elements is used, but no code is written into the header section, ASCET doesn't generate a header file when “Transfer To LCO”. Unfortunately the generated SCOOP-IX file contains a reference to this non-existing header-file. Due to this the build in LCO fails.

Workaround:

Introduce a comment into the header section. Now ASCET generates also a header file.

ASCET V6.1.1 or higher: New entry "Transfer to LABCAR" in combo box "Select Online Type" for target RTPC

Starting with ASCET V6.1.1 + LCO V5.0.2 there is an extra Online Target "LABCAR" for code transfer to LCO:



Please use this entry for transfer to LCO (target RTPC) and not the entry INTECRIO anymore.

ASCET >V6.0.1 does not offer hierarchical names for global elements

With previous versions of ASCET it was possible to determine by the setting "Hierarchical Names" whether the label for global elements (parameters, measurements, messages) should reflect the area of validity (scope) or the position of definition in the ASCET models hierarchical structure.

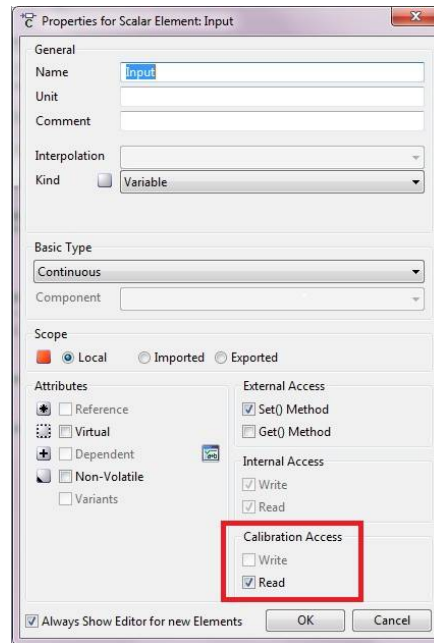
Starting with ASCET V6.1.0 all global elements are always placed in the top level of the ASCET project. In the Workspace Elements tree of Experiment Environment they are arranged in the folder drawn for the ASCET Module regardless they are defined (exported) on a lower hierarchical level in the ASCET project. If you upgrade an ASCET project with setting "Hierarchical names" to a newer ASCET version and import it in LCO you will need to re-establish connections and adapt the parameter files according to this change.

ASCET >V6.2.0HF1: CT block method "UpdateDependentParameters" not triggered for changes if parameters imported from SD components

If a global parameter defined in an SD component (Module, Class etc.) as exported and imported into an CT Block, its change will not trigger the execution of method "UpdateDependentParameters" and therefore parameters depend on it will not get updated. Only parameters defined in CT blocks (local, or imported from other CT blocks) are monitored by the CT block state machine and will schedule the execution of method DependentParameters on change.

ASCET V6.2.0: Correct Update of Dependent Parameters requires property CALIBRATION ACCESS READ set

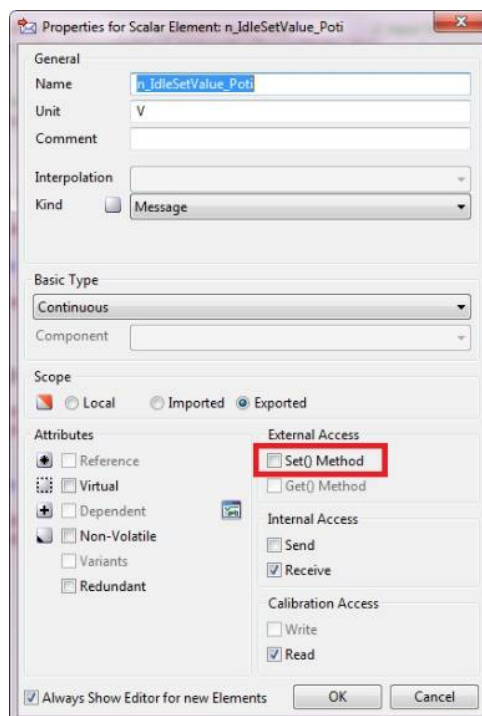
To ensure dependent parameters are created with all necessary attributes in the generated C code make sure having the property CALIBRATION ACCESS – READ activated in the elements properties dialog (see the following screenshot):



ASCET V6.2.1: A label which was generated as LCO-Inport with ASCET V6.2.0 is only available as measurement within ASCET V6.2.1 (EHI # 389979)

In the past a element of type 'Message' was handled only via the 'Internal Access' properties. Was the internal access set to RECEIVE only the 'Message' was handled as a RECEIVE only without consideration of the 'External Access' properties. These issue was fixed with ASCET V6.2.1 since also the external access can change the 'Message' behaviour. If you specify the external access as Set() and the internal as RECEIVE the message itself is handled as SEND/RECEIVE item. If you specify the "ignore internally connected elements" option within LABCAR-ASC such an element will be ignored too while generating the LABCAR elements.

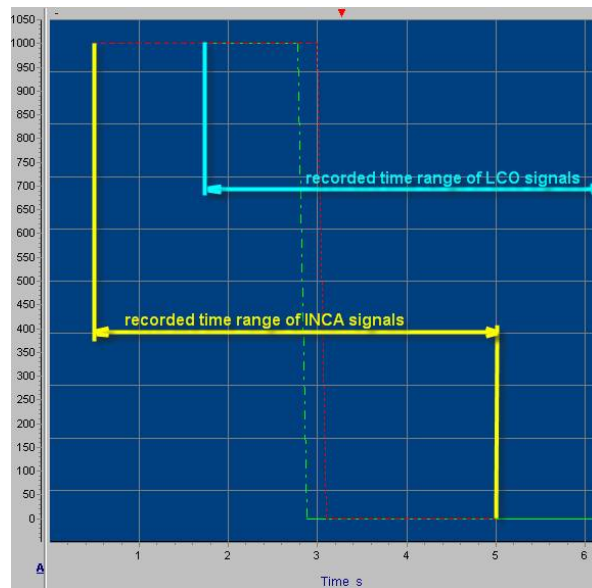
To fix this issue you should remove the Set() method option from such 'Message' elements.



INCA[®] related issues

Jitter in recording frame when using LCO and INCA Dataloggers in parallel and an INCA signal is used to trigger the recording

If an INCA signal is used to derive a start and/or stop trigger condition a non-real-time mechanism is used to determine the state of the INCA Datalogger when starting the data transfer to file for permanent storage. This leads to a difference in the start and stop of recording and results in a shifted recording frame for LCO and INCA signals (see picture below):

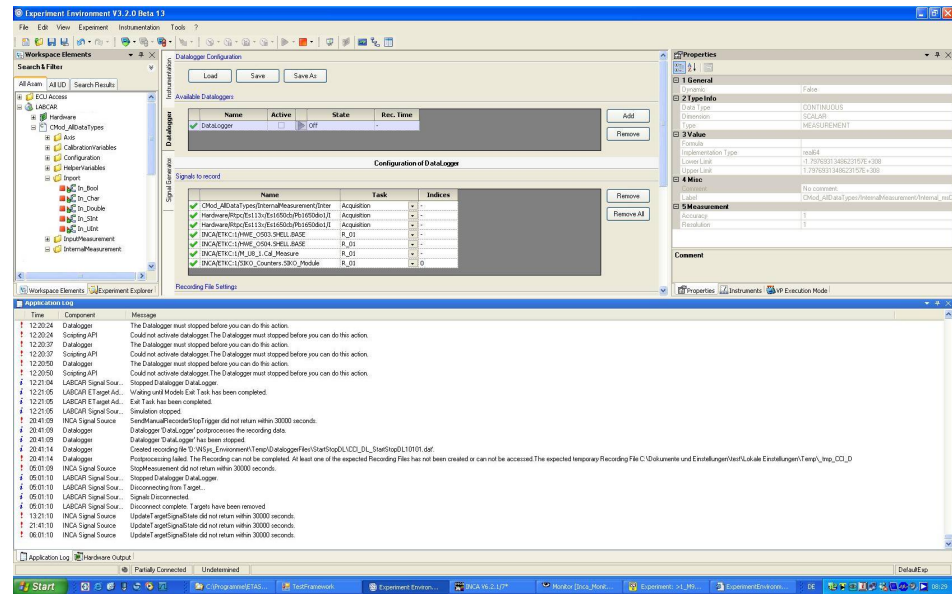


To avoid an unhappy cut off in the displayed time range, the pre and post-trigger times should be selected appropriately (enhance the range of interest by approximately 2s).

In any case, a post-processing of recorded data is done to synchronize the time scales; so there is no time shift in the time axes of both devices.

Sporadic problem with heavy duty mixed LABCAR-OPERATOR V5.4.0 and INCA V7.0.0 Datalogger operation (EHI # 276200)

During a LABCAR-OPERATOR beta test the following problem was observed: When starting and stopping the Datalogger more than 100 times within 4 s and mixed device INCA and LCO signals are assigned to be logged the following sporadic error occurred "INCA Signal Source: [name of API function call return value] did not return within 30000s" (see picture below).



Problem is solved if you use INCA V7.1.8 or higher instead.

Oscilloscope does not run after an INCA measurement is added

If fast signals from RTPC and INCA are measured in the same oscilloscope, and there are large differences in the measure rates from these two sources, the oscilloscope may hang. The work-around is either to sample the RTPC (or E-target) signals in a slower raster, or measure the INCA values in a faster raster.

INCA labels already present in INCA experiment are not recorded when only monitoring them with LCO-EE (EHI # 315039)

The following problem refers to remote operation of INCA with LCO-CCI: When opening the INCA experiment with LBCAR-OPERATOR, measurement elements may be already present in the INCA Experiment Environment. When selecting the same measurement element again to be displayed in the LCO-EE, the STATE property in the INCA EE is switched from "Disp/Rec" to "Disp. only". After that these elements will not be included in an INCA Datalogger when performing a recording session in INCA without engaging the Datalogger in LCO-EE.

You can either change the state back again in the INCA Variable Configure dialogue or use the LCO-EE Datalogger to perform recordings of INCA variables.

Walkback and a system error window appear during automated test in INCA (EHI # 350391)

The following issue had been observed during our release tests on WINDOWS 7 with INCA V7.0.0: In rare situations during automated test run a system error occurred in case the mouse pointer is placed over a user interface control with a tool tip attached. Detailed information regarding circumstances is not known so that a fix cannot be provided yet. As a work around we recommend to place the mouse pointer at the border of the screen while an automated test is running.

Issues related to Module for Realtime ECU Access XETK via XCP on Ethernet (LCO AddOn LCO-LCX)

Only hook based bypassed quantities may be selected as LABCAR-Inport

At the moment only quantities with hook based bypass may be stimulated by assigning them to a LABCAR-Inport.

Required information in A2L file used for XCP Module

The A2L file imported as configuration file for the Module need to provide values for communication parameters for the XCP connection (section "IF_DATA_XML" in A2L file) and a section describing the interface structure (section "ASAP2 meta language (AML) for XCP V1.0.1").

Access to XETK via LCO-LCX and INCA

When accessing the XETK device via INCA and LCO in parallel (multi master operation), it is not possible to select the same rasters for both devices. Only the first master will get access, for the second master the error message "raster not available" will be issued (for LCO this message is displayed in the RTPC log file).

Discontinuation of LCO V5.0.x XCP Module

With LCO V5.1.0 the new common ECU Module for integration of XETK as well as ETK as digital ECU interface into an LCO project have been introduced. Together with that the previous "XCPModule" have been discontinued. Please take into consideration that there is no migration from "XCPModule" to "ECUModule": Instead the "XCPModule" need to be removed from the LCO V5.0.x project **before** migrating it to a newer LCO version. Otherwise build errors will occur when trying to build the migrated project:

```
Enter: ./simodels/temperatureconversion/objects/src
hs-uninstall:
mc_module_xcpmodule.c: In function 'initClass_Module_XCPModule':
hs-clean:libsmodels_temperatureconversion_objects.a
mc_module_xcpmodule.c:27: error: 'XCPModule_MeasStruct' undeclared (first use in this function)
mc_module_xcpmodule.c:27: error: (Each undeclared identifier is reported only once
mc_module_xcpmodule.c:27: error: for each function it appears in.)
make[3]: *** [../obj/mc_module_xcpmodule.o] Error 1
make[3]: *** Waiting for unfinished jobs....
Leave: ./simodels/temperatureconversion/objects/src
Leave: ./simodels/temperatureconversion/objects
```

Issues related to Module for Realtime ECU Access via XCP on CAN (LCO AddOn LCO-LCX)

Limited band with for monitoring measurements

At the moment the size of data acquisition for measurements is limited to 8 byte measurement raster (1 DAQ list per raster).

Issues related to Module for Realtime ECU Access ETK via ES592 or ES910 (LCO AddOn LCO-LCE)

Change of parameters with ETK

The change of parameters is only supported for scalar parameters (no maps/curves) for

parallel ETK. It is required that the Working Page is set as active page.

If INCA is connected at the same time the changed parameter value is neither transferred to the INCA experiment environment nor indicated as changed.

Grace Mode warning after pushing the FiL START COMMUNICATION button

The first time the START COMMUNICATION button is pressed after installation of LCO a grace mode warning like the following may be displayed, even if the License Feature "INT_UCC_LCO_FIL" is granted:



Just activate the checkbox "Never show this dialog again" and confirm the dialog with OK. For future version we work on a solution which does not show this dialog.

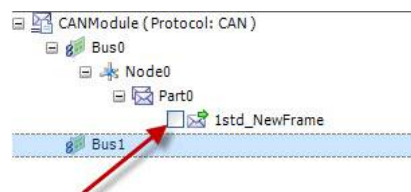
Access to ETK via LCO-LCE and INCA

When accessing the ETK device via INCA and LCO in parallel (multi master operation), it is not possible to select the same rasters for both devices. Only the first master will get access, for the second master the error message "raster not available" will be issued (for LCO this message is displayed in the RTPC log file).

Issues related to CAN Module

CAN code generation problem when deselected or remove a CAN message with user defined code assigned (EHI # 318162)

When uncheck check box "Enable/Disable Message for Code Generation" in LCO-IP CAN Editor tab, the message is also not part of header files that define the CAN rest bus message data structure. The same is true in case the message or a complete part is deleted.



This may lead to build errors in case the message is still part of User Code assigned to any of the remaining CAN messages. The user code needs to be adapted in a way the structure elements of the deactivated CAN messages is not referenced in any way to make the

compilation running successful.

Change of labels for CAN Signals in stored connection files (LCD-files)

The hierarchy of labels displayed in the Connection Managers Input and Output section has changed. All connections are migrated on project migration from LCO V5.x.y to new LCO. Nevertheless if you want to re-use connection files stored with a previous version of LCO you have to adapt the hierarchy in the label names in the following way:

old Path string:

```
...<Path>TestDBC/SendMessages/7bstd_Test_123/int2unsig</Path>...
```

Or

```
...<Path>TestDBC/SendMessages/Board_0/Controller_0/7bstd_Test_123/int2unsig</Path>...
```

New path string:

```
...<Path>[module name]/Bus_0/[node name]/7bstd_Test_123/int2unsig</Path>...
```

The exact string to be added depends on the number of busses and nodes defined for the virtual network to be simulated. This can be determined by the search function in Connection Manager or by checking the file "Connections.lcd" created for the migrated project during the migration.

Receive CAN message lost if receive task is triggered too slow

In case of the receive messages rate is higher than the scheduled period for receive task the situation may occur that data get lost due to of buffer overflow. You may identify this situation by the following line in hardware log output

```
<ixxat-can>ixxat_can:0-0. CAN receive messages lost (100)!  
Restarting CAN BCI is in wrong state'
```

With LABCAR NIC you can receive even messages that are not scheduled in CAN Editor during edit time. Therefore this situation cannot avoided by calculating another period for the task the receive process is assigned. You can fix the issue of drop messages by separating receive and send processes to different tasks and/or increase the task rate for the receive process to the fastest period of receive signal.

Restrictions of support J1939 CAN Layer Extensions by LABCAR-NIC

Starting with version V5.0.0 LABCAR-NIC supports CAN messages with more than 8 byte payload according the J1939 specification. With V5.2.0 this support has been enhanced. Nevertheless currently this addresses only enhancements for Data Link Layer according J1939-21 (Parameter Group Numbers, Source Addresses, PDUs) and for Network Management according J1939-81 Section 3.2.1 (Non-Configurable Source Address). All other parts of J1939 are not supported.

Restrictions that apply to the SAE J1939-81 CAN extension in LCO V5.2.0

1. Currently you can configure a fix source address only. This corresponds to the section 3.2.1 "NON-CONFIGURABLE ADDRESS ECU" of the SAE J1939-81 standard. Though the address may be modified during runtime it cannot be changed by the network system itself. This means the 3.2.2 SERVICE CONFIGURABLE ADDRESS ECU, 3.2.3 COMMAND CONFIGURABLE ADDRESS and 3.2.4 SELF-CONFIGURABLE ADDRESS are not supported.
2. Section 4.4.1.
ADDRESS CLAIM REQUIREMENTS—Every ECU is required to claim its source address upon initialization and upon any change of the ECUs NAME or source address. Even though the user is allowed to change the NAME data (via Inport to Address Claim message signals), we do not automatically trigger the transmission of the Address Claim message when any of these signals change. The user may install a state machine so that a new Address Claim message is sent.
3. Section 4.4.2.3
There is no response to Address Claims of Own Address—An ECU should transmit an Address Claim if it receives an Address Claim with a source address that matches its own, and if its own NAME is of a lower value than the NAME in the claim it received. If the ECUs NAME is of a higher value than the NAME in the claim it received, the ECU should not continue to use that address. (It may send a Cannot Claim Address message or it may attempt to claim a different address.).
4. Section 4.4.2.4
An ECU previously listening to this ECU should detect that the ECU has become disabled or changed its address by monitoring the Address Claim by the more dominant ECU (lower name value) as well as by monitoring the Cannot Claim Address message. LABCAR itself takes no action. It is up to the user to configure UuT Nodes to receive messages from any SA that a real ECU may claim.
5. Section 4.4.3.3
After transmitting any claim message, the transmitting ECU should monitor error code information. If the error code indicates that a bus error has occurred, any automatic retransmission attempts by the CAN peripheral should be canceled if possible. We do nothing special on failure to transmit. It is possible to get the transmission status (failed or succeeded) of a message, but nothing is done with this status.
6. Section 4.4.3.4
The ECU should respond to a request for Address Claimed with a Cannot Claim Address message after a transmit delay (see 4.4.3.3). In the case where there is a collision of the Cannot Claim Address messages, the process in 4.4.3.3 should be used. In the current implementation a Cannot Claim message is sent, but we do introduce a pseudo-random delay. Further there is no attempt to retransmit if transmit fails.
7. Section 4.5.2.4
Continuity of Addresses Across Power-Down and Power-Up Cycles—ECUs should be able to maintain their source address and any addresses for ECUs that are communicated with so that the ECU can attempt to use the same addresses at the next power-up. This should be done except in cases where special requirements override this recommendation, for example, in towed subnetworks of Off-Highway trailers where the instance and associated addresses of the trailer may change at each power-up. However, there is no continuity of addresses across Download/Disconnect in LCO-EE.

LABCAR-IP memory consumption increases during import of AUTOSAR System Template database files (*.arxml) for CAN

During import of AUTOSAR *.arxml files the memory consumption of LABCAR-IP increases. Therefore it might be necessary to close and open LABCAR-IP when working with large (> 10MB) *.arxml files.

To overcome this issue we recommend considering the following points:

- Switch off the automatic reload of database files
- To avoid data loss, perform intermediate save actions when importing and working with *.arxml files (before and after import as well as when dragging parts of the imported file to your bus definition).

In addition we recommend to avoid reloading such CAN modules (double-click on module within project explorer) all over again to avoid continuous increasing of memory usage!

J1939 Cluster within AUTOSAR System Template database files (*.arxml) currently not supported

During import of AUTOSAR *.arxml files the specific J1939 clusters are currently not recognized from the arxml parser. This will be improved in one of the upcoming LABCAR-OPERATOR versions.

Terminating resistors permanently needed for XC16, IB200 and IB600 board if STT feature is used

If the STT (Single Transmission Try) feature is used within CAN configuration the terminating resistors need to be in place at any time to avoid communication malfunctions on the CAN bus.

Also a temporary removal of the resistors needs to be avoided strictly.

LABCAR-IP memory consumption increases during re-opening CAN Module editor several times if AUTOSAR System Template database files (*.arxml) are used

If you re-open the CAN module editor over and over again and your module contains linked huge AUTOSAR System Template database files it may happen that LABCAR-IP consumes more and more of the available application memory.

We currently work on a solution for one of the next LABCAR-OPERATOR versions but until this issue persists we recommend to avoid such re-opening scenarios.

Issues related to FlexRay™ Module

FlexRay™: Send period for FlexRay™ message need to be set in EB tresos® Busmirror® explicitly

In Fibex standard the send period of a message is determined implicitly by the slots a FIBEX Message is assigned. There should not be an explicit cycle time set because this would be an over specification which may lead to problems in case both values turn out to be different. With EB tresos® Busmirror® instead it is required to specify an explicit cycle time. This is to enable a maximum flexibility in case frames should be used for cyclic and event timings.

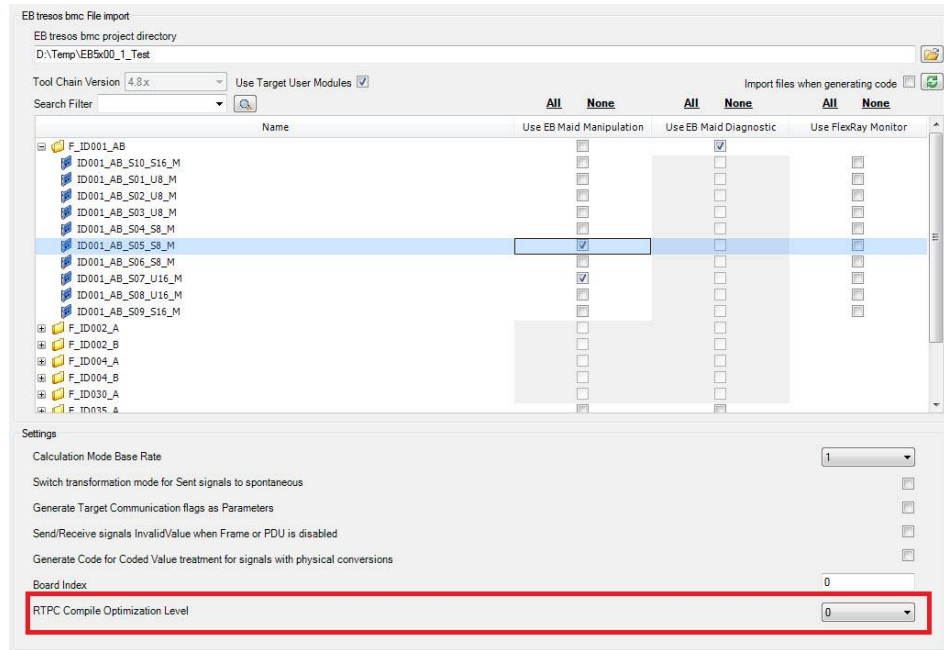
FlexRay™: Signals with transport data type of 64 bit length not supported

The current version of EB tresos® solution of FlexRay™ residual bus simulation only supports a maximum signal length of 32 bit. Therefore the maximum bit size of binary coded signal transferred with one TAL call of "TAL_BM_SendSignalCoded" or vice versa "TAL_BM_ReceiveSignalCoded WithStatus" must not exceed 32 bit corresponding to the setting "Transport Data Type" in EB tresos® Busmirror®. Otherwise a cutoff will occur. If a data pattern larger than 32 bit needs to be transferred it needs to be split among different signals.

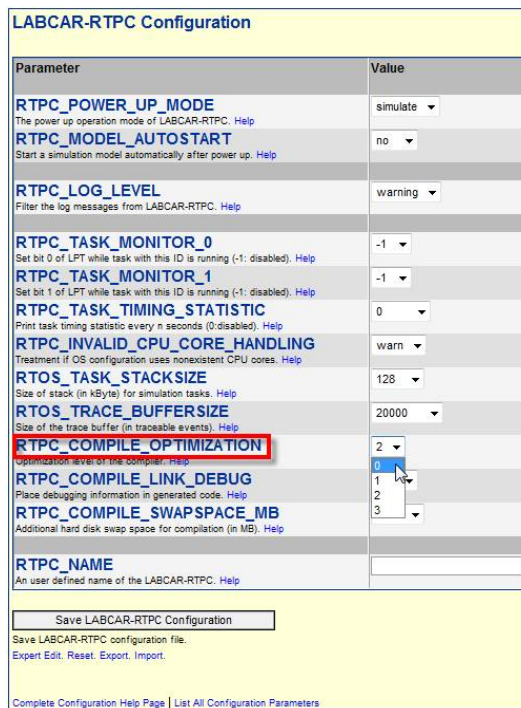
FlexRay™: Error "cc1: out of memory allocating 319034700 bytes after a total of 931389440 bytes" on compilation of file "flexraycode.c"

With large FlexRay™ residual bus simulation projects the compiler error shown above may cause by memory exhaustion during compiler optimization. We work on a solution for this issue. As a work around we recommend the following: Reduce the compiler optimization level in the RTPC Web Interface on page Configuration.

This may be achieved by setting the corresponding parameter in the FlexRay™ module user interface.



Alternatively the setting may be changed in the RTPC Web interface for the complete project:



FlexRay™ Module: Operation of sent frames in LABCAR-Bus Communication Monitor (LABCAR-BCM)

A calibration of signals assigned to send frames is only possible if the frame is just sent by the EB tresos® Busmirror® simulated ECUs. If the frame is sent and received by Busmirror® simulated ECUs the corresponding signal values and properties cannot be changed in LABCAR-BCM.

FlexRay™ Module: How to remove elements folder EB MAID quantities

The following steps should be performed to remove the folder for EB MAID quantities in case they are no longer required:

- a. In EB tresos® Busmirror®, remove the manipulation TUM assignments Module Configuration and rebuild application.
- b. Deactivate option „Use EB MAID Target User Modules“
- c. Activate button „Copy Files Again“
- d. Save NIF-Modul and build LCO project

FlexRay™ Module: Drop outs of FlexRay™ data received on RTPC

The send/receive process of a FlexRay™ module not only processes the data transfer between FlexRay™ hardware and RTPC but also triggers additional functions such read of slot buffer for received data, the error channel queue and the transfer to the TAL client interface (called "TAL upper half").

If triggering these activities is slower than the production of new data (e.g. received frames/PDUs or error messages) an overflow of buffers may cause a loss of information. Therefore a) a high reception rate (e.g. event triggered data from dynamic slots) b) a higher amount of error messages or c) a too low period of FlexRay™ processing task on RTPC may lead to the situation the data for receive signals on RTPC remain unchanged (drop out or stall) until new data may be transferred again after the fix of the overflow situation.

To avoid such a situation the period of the task the send/receive process of FlexRay™ module is assigned (default: "Module_[module name]_SendReceive") should set appropriate. You can adjust this period by changing the parameter "Calculation Mode Base Rate" in the FlexRay™ Editor window. The higher the number the smaller is the task period.

We recommend to select a period that fits to the rate of the fastest receive signal.

FlexRay™ Module: Activation of Status logging via serial line interface on EB5200

When connecting the serial interface of EB5100 to terminal program status information regarding the hardware status can be monitored. With EB5200 this monitoring need to be activated in the Busmirror® software explicitly: Change the line

```
#undef EBX2XX_ENABLE_ALIVE_PRINTF
```

```
At the end of the file "...\\tresos_Busmirror\\4.6.1\\tresos\\plugins\\Base\\src\\TUM_IdleTask.c" to
```

```
#define EBX2XX_ENABLE_ALIVE_PRINTF
```

to activate the serial status logging for EB5200.

FlexRay™ Module: Restrictions for Signal Groups

The following restrictions for signal groups in the FlexRay™ configuration apply:

1. Signal Groups must be subsets of PDUs, e.g do not distribute among several PDUs.

2. Signal Groups must occupy discontinuous bits in the PDU bit pattern; there cannot be spaces between bits that belong to the same or different signals.
3. Signals can only be once in a Signal Group.

FlexRay™ Module: Change in processing invalid values for bool signals

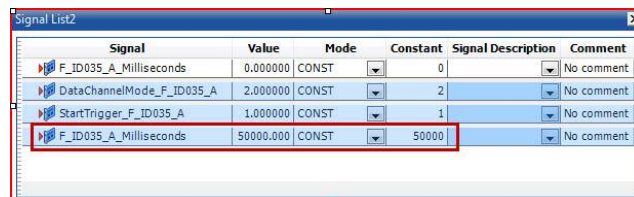
Due to changes in EB tresos® Busmirror® TAL >=V4.4 exception handling the processing when exchange signals of type Boolean from RTPC to EB5x00 need to be changed as follows:

- a) LABCAR-NIF in LCO <V5.2.0: If a value greater than 1 was mapped to a bool signal the value was limited and a TRUE was taken as actual value to be transmitted. (consider values for LABCAR ports are implemented always as IEEE 754 double-precision binary floating-point format).
- b) LABCAR-NIF from LCO V5.2.0 on: If a value greater than 1 is mapped to a bool signal the value is considered as invalid and neglected. Instead the previous value – which can be either TRUE or FALSE remains active and will be transmitted regardless the definition of INVALID value of the signal in the original FIBEX configuration is.

FlexRay™ Module: Define pause timing for PDU Manipulation

PDU manipulation enables to pause the transmission of PDU signals for a certain time. However the defined duration values are used incorrectly (divided by the factor '10') for values > 4.999ms.

Therefore this requires taking this into account when defining those pause timings. For example for a desired pause time of 5sec (5.000ms) you have to define 50.000ms.



Signal	Value	Mode	Constant	Signal Description	Comment
F_ID035_A_Milliseconds	0.000000	CONST	0		No comment
DataChannelMode_F_ID035_A	2.000000	CONST	2		No comment
StartTrigger_F_ID035_A	1.000000	CONST	1		No comment
F_ID035_A_Milliseconds	50000.000	CONST	50000		No comment

However for smaller values (<= 4.999ms) like 1sec (1.000ms) this must not be done. Here 1.000ms would be the correct value to use.

Issues related to other parts

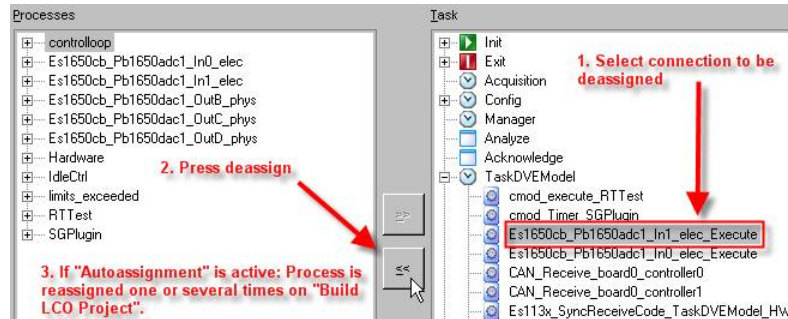
Conversion Module process assignment after change of connection

When changing a connection of type "OLC Conversion" in LCO-IP Connection Manager the process of the corresponding Signal Conversion Module needs to be reassigned. This is required either

- a) if an established connection is broken. In this case the process for signal transformation calculation "[Signal Conversion Module Name]_execute" need to be re-assigned.
- or
- b) if the output of the module is to be assigned to an additional LABCAR Inport calculated in another task. In this case the process for signal transformation

calculation “[Signal Conversion Module Name]_execute” need to be assigned once more.

To trigger a reassignment, select the affected process “[Signal Conversion Module Name]_execute” in the OS Editor of LCO-IP and press the button “Unassign”



Take notice that an existing Conversion Module cannot be “reused” when establishing a port connection of type “OLC Conversion”. In this case always a new Conversion Module is inserted. To “reuse” a Conversion Module just establish a connection without conversion to the port of the existing Conversion Module.

Re-import LDF files or Parts in LIN (EHI # 324425)

In contradiction to CAN when re-import a LDF file or a Part no merge dialog is displayed. Instead always a new item is created for the imported Part or LDF file. We work on this topic with the intention to provide a merge dialog also for the LIN Module.

LABCAR-IP hangs showing “No Response” during build process (EHI # 379212)

The current build process runs together with the UI thread. Some build steps take so much time that LABCAR-IP seems to hang or crashed. If you have a look within the task manager you can observe that the process is fully working. It is highly recommended to keep this running and not to kill the process. We will come up with a better solution in one of our next versions.

Error within RTIO table editor of ES1335 configuration (EHI # 396634)

The short-cuts Ctrl-A, Ctrl-C and Ctrl-V are not functioning properly. Please use the export to file, import from file functionality instead.

Issues related to the Experiment Environment

3D arrays are not supported

3D arrays are not supported; we recommend using a number of 2d arrays instead.

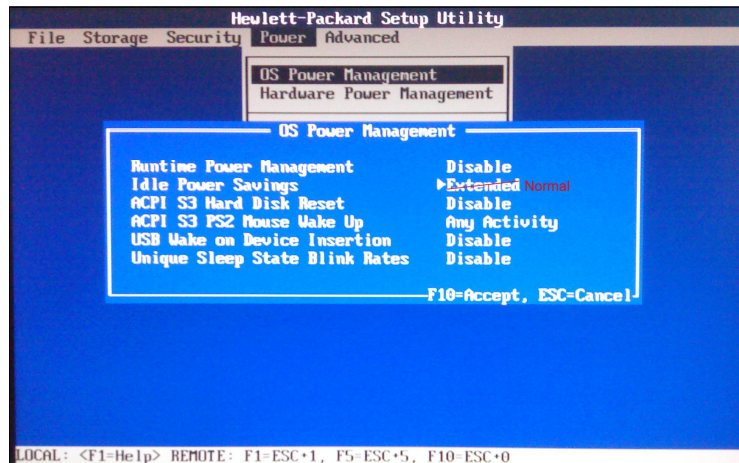
Special method syntax for Python scripting

With Python as scripting language, argument less methods must be used without parentheses for the automation of the Experiment Environment. E.g., `SignalSources.Download()` will fail.

Recommendation: use `SignalSources.Download` instead.

Data acquisition problems when PC power management is active

On certain PCs problems with data acquisition have been observed in case the CPU power management is set to active in the PCs BIOS settings. As an example the Compaq Elite 8000 Convertible Mini Tower BIOS settings are shown in the following picture:



The problem occurs in case the property "Idle Power Savings" is set to "Extended". If the setting is "Normal" the problem was not observed. In addition to that the problem seems to be suspended in case any device is inserted to any of the USB sockets of the PC.

During the simulation in LABCAR-EE the data acquisition stops periodically for a certain time which causes leaps in the oscilloscope and in the recorded data.

We recommend disabling all power management settings in case your PC hardware shows the behavior described.

Problem downloading arrays

If arrays are downloaded to a target, and the array is larger in the Experiment than in the target model, then the data is downloaded anyway without any warning. This is incorrect. The user should be warned there is a difference in array sizes, and the download interrupted.

Performance problem when using a large number (several hundreds) of instruments of type "Edit Box"

Please be aware that the display instrument "Edit Box" is very resources consuming in reference to MICROSOFT Windows™ GDI objects. If you want to create Experiment Environment layers with a large number of display elements we recommend to use the 7 Segment Display which does not affect the number of GDI objects used for the Experimental Environment.

Datalogging with tasks of type "Event" as acquisition task

If the option the AUTOTRIGGER is active, the lower limit of task activations (process raster) is 2ms. If you enter a value smaller as 2ms no values are acquired and recorded. If recordings with smaller period need to be performed use a timer task instead.

Signal Generator: Tasks of type “Event” are not supported as task for stimulation

Tasks of type “Event” cannot be used to trigger data generation in the EE Signal Generator.

New platform Signal Generator does not support signal type Function Signal

With signal specification of previous Signal Generator (LCO <V5.1.0) (Stimuli Set) it was possible to define intervals of type Function Signals for a Channel. The Function Signal is to generate a stimulus as a mathematical function of time (like a function generator), e.g. $4 \cdot \sin(5 \cdot t) \cdot \exp(3 \cdot t)$. We consider offering this type of signal in a succession version of the new Signal Generator.

New platform Signal Generator trigger implicit stop on re-configuration

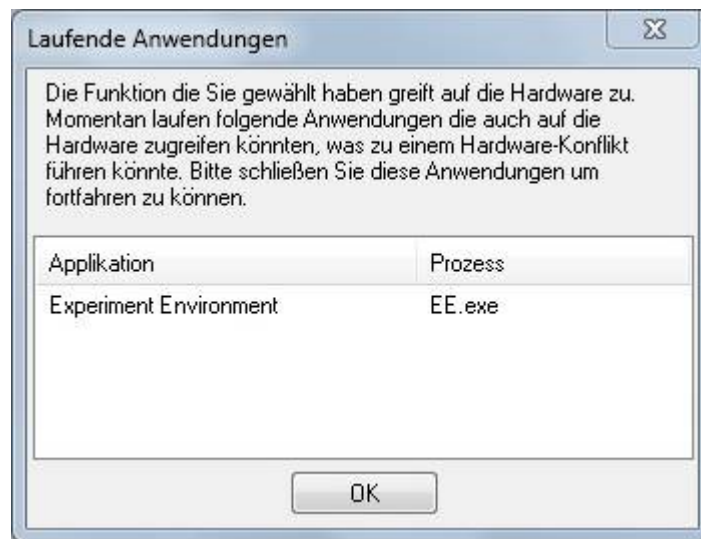
To provide compatibility with earlier API versions of EE (<V3.3.0) it is possible to re-use scripts on basis of the previous Signal Generator API even with the new Signal Generator. Nevertheless one adaption need to be applied to accomplish the new state machine of the Signal Generator: In case a running Signal Generator is to be re-configured (change of setting for one of the properties START VALUE, STOP BEHAVIOR, SIGNAL DESCRIPTION SET, DEVICE and TASK) an explicit stop of the Signal Generator need to precede the reconfiguration request. The new Signal Generator does not trigger an implicit stop in case a reconfiguration request is raised.

New platform Signal Generator: Signal Descriptions with wave form “Random” and “Data” are not stored in STI format

If you store a SignalDescription in STI format you will get the error “SignalDescription '[NAME]' could not be exported: 'Random' or 'Data' wave forms are not supported in sti format!” This is because of the STI format as defined by the ASAM AE HIL API standard does not support these wave form types in the current used version. We currently work on an update of our STI format importer/exporter. Please store the Signal Description in ESTI format instead for now.

Parallel operation with INCA: Warning dialog if “automatically update of hardware” is active

If LCO and INCA are used in parallel the following window may appear in case the INCA option “Automatically Update of Hardware” is active:



We recommend switching the option "Automatically Update of Hardware" off in favour of a well-defined tool version setup during test execution.

Known issues HiL API Support

The following functionality defined in the ASAM AE HiL API V1.0.2 for ECU Testing via Hardware-in-the-Loop Simulation is currently not supported by LABCAR-OPERATOR:

1. Manage Signal Description via HiL API: LCO-EE provides methods to process segments in a signal description but with its own proprietary library. The API methods from the interfaces of the Standard HiL API are not supported.
2. The access to a Signal Description Set file (STI) via API is not supported.
3. The MDF 4 file format is not supported yet; only MDF 3.0 files may be imported for Signal Generation and stored by Data Logger.
4. It is not possible to apply down sampling to captured data.
5. The functionality "memory writer" during data capturing is not supported.
6. The access to recorded data while recording (online data) is not supported.
7. It is not possible to get the current value for the stimulation time (time elapsed).
8. The ports "ECU Access", "Diagnostics" and "Access to Error Simulation Facility" are not supported by LABCAR-OPERATOR.

Known issues XiL API Support

The following functionality defined in the ASAM AE XiL API V2.0.1 for ECU Testing via Hardware-in-the-Loop Simulation is currently not supported by LABCAR-OPERATOR:

1. Manage Signal Description via HiL API: LCO-EE provides methods to process segments in a signal description but with its own proprietary library. The API methods from the interfaces of the Standard HiL API are not supported.
2. The access to a Signal Description Set file (STI) via API is not supported.
3. The MDF 4 file format is not supported yet; only MDF 3.0 files may be imported for Signal Generation and stored by Data Logger.
4. It is not possible to apply down sampling to captured data.
5. The functionality "memory writer" during data capturing is not supported.
6. The access to recorded data while recording (online data) is not supported.
7. It is not possible to get the current value for the stimulation time (time elapsed).
8. The ports "ECU Access Calibration", "ECU Access Measurement", "Diagnostics", "Network" and "Electrical Error Simulation" are not supported by LABCAR-OPERATOR.

Known issues when opening SUT Mapping File Editor Online Help the first time (EHI # 351026)

When you try to open the online help directly after installation of LCO from the help menu of the SUT Mapping File Editor you may receive the following error window:



An easy work around is possible: Just execute the program C:\Program Files\ETAS\LABCAR-SMFEEditor 4.0.1\SMFEhelp.exe for one time manually. On the first start of the help viewer the extension ".mchelp" is registered to the help viewer.

Runtime error messages in case multiple instances of same kind of LCO Behavior Module used in LCO project

With LCO V5.2.0 it is possible to have multiple CAN- LIN- and FlexRay™-Modules in one LCO project. Nevertheless currently there is no cross module consistency check. The user needs to take care that the names of shared elements are unambiguous and unique. This affects especially the IDs for hardware controllers, assigned to the Behavior Module in the corresponding editors. Example: When assigning a CAN controller twice to 2 different Behavior Modules the following error message is displayed in the hardware tab of the LCO-EE's log view:

```
Error 10:28:42 RTPC Syslog <ixxat-can>ixxat_can_open: Already connected to controller 1
Error 10:28:42 RTPC Syslog rtos_comm_open failed
for:ixxat_can:0/controller=1 btr0=0 btr1=28 stt=0 checkinterval=0.5
initchk=4
Error 10:28:42 RTPC Syslog Error in initializing CAN card for bus 'Bus_1' with CAN-controller 1 of CAN-board 0.
Error 10:28:42 RTPC Syslog rtos_comm_read: Communication channel not open
```

EE API Method GetLiveImpIValueObject does not return the binary coded INCA value (EHI # 328285)

With LCO V5.2.0 (EE version 3.4.0) it is currently not possible to receive the implemented (raw) value of an INCA quantity. If the method is called, always the physical value is returned. We will extend the capability within the next EE versions.

EE Table Editor Instrument does not support ENUMs (EHI # 336261)

Currently the EE Standard Instrument "Table Editor" does not support ENUM data type elements. We will extend the capability within the next EE versions.

4.2.3 Hardware related items

Position of the ES1130 within the ES4xxx housings

Please be aware that the position of the ES1130 within the ES4xxx housing has to be carefully considered:

- ES41xx: The ES1130 has to be the outer left card of all (not necessarily in slot 0)
- ES4205: The ES1130 has to be in slot 0
- ES4300: The ES1130 has to be in slot 0

Multiple VME systems with one RTPC

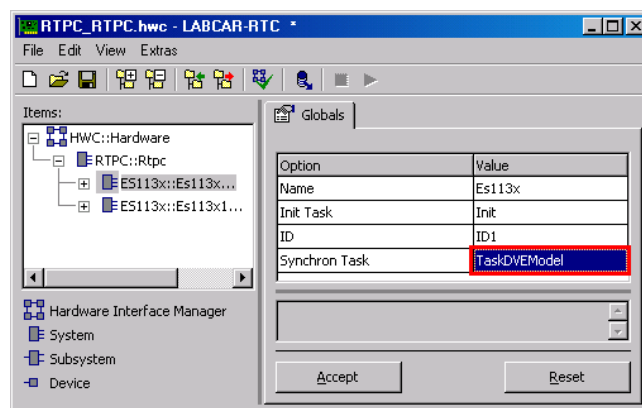
The assignment of processes needs to be rearranged when more than one ES1130 VME housing is to be integrated into a LCO project (so called Massive-IO). For efficient data transfer the Synchronic Transfer Mode between VME-Bus controller ES1130 and the Real-time PC need to be activated. In this mode 2 additional processes for hardware control are assigned to the task selected for to the property SYNCHRON TASK on the GLOBAL tab of every RTIO sub system of type "Es113x". If more than one VME system is used several processes "ES113x1_Sync...", "ES113x2_Sync..." etc. are created and assigned.

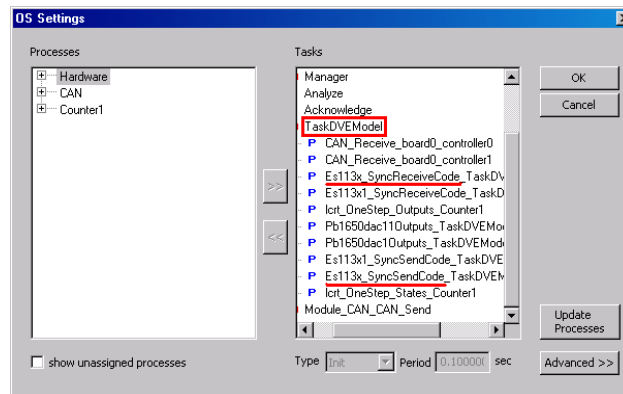
During setup of a LCO project with more than one VME hardware systems it may happen that the task assignment of the different ES113x_SyncSendCode processes becomes incorrect. This will lead to a break down in the communication between the VME-hardware and RTPC with an error message being reported after START EXPERIMENT and subsequent completion of the hardware initialization phase:

```
communication time violation RTPC <--> ES1130-1 (0.620136s)
rrtio synchron(1): Invalid number of data bytes (66/62)
```

To solve the problem perform the following steps while LCO experiment is closed:

- 1) In LCO-IP: Activate tab OS Settings,
- 2) Unassign ALL (!) processes on ALL tasks by selecting them and pressing the "<<" button.
- 3) Perform an auto-attach on all unassigned processes.
- 4) Verify, in the process assignment for the SYNCHRON TASK that the ES113xn_SyncSendCode... processes are the last ones in the process list and also are positioned before lcrOneStep_States_vehiclemodel. See the following pictures:





The error messages do not appear again after the project is rebuilt.

FlexRay™ Module: Same EB tresos® Busmirror® version should be used for all modules

Starting with LCO V5.1.0 it is possible to set up several EB5100 FlexRay™ boards with one EB tresos® Busmirror® project each. Nevertheless all projects need to be built with the same EB tresos® Busmirror® version. It is not supported to mix EB tresos® Busmirror® software versions.

EB5100 and EB5200: Downdate of firmware is not supported

The Elektrobit hardware EB5x00 can be used with the most recent version of FPGA image and Operating System even if you use different tool versions of EB tresos® Busmirror. You can use the most recent version provided with HSP even if you work with previous/different versions of the Busmirror tool. Therefore the downdate of EB5x00 firmware is not supported.

4.3 Compatibility issues to projects created with earlier versions of LABCAR-OPERATOR

Limitations in migration of single project components

With LABCAR-OPERATOR it is in general not possible to re-use single files generated by the tool e.g. Behavior Module files (file extension lmd), configuration files (file extension lvd) in projects of different LCO versions. These files need to be processed by the LCO Migration Wizard of the newer version in the original project context to perform automatic migration successfully.

Build problems with VDYM V5.0.x when using with LCO V3.2.3 (and higher) and RTPC V3.0.0 (and higher) (EHI # 314616 and # 314617)

To operate VDYM V5.0.x with LCO V3.2.3 (and higher) and RTPC V3.0.0 (and higher) the following manual adaptations are necessary to finish the build process successfully. During build, the following error message is displayed in the HTML BUILD Log window with the unchanged LCO project:

```
Compile: f_compsemitrail_a_v1_0_1_compsimpacksemitrailingvehicle.c
f_compsemitrail_a_v1_0_1_compsimpacksemitrailingvehicle.c: In function 'mdlInitializeSampleTimes':
f_compsemitrail_a_v1_0_1_compsimpacksemitrailingvehicle.c:3322: error: invalid lvalue in unary '!'
make[5]: *** [../obj/f_compsemitrail_a_v1_0_1_compsimpacksemitrailingvehicle.o] Error 1
```

Depending on the model variant used, one of the following C code S-Funtions will be affected:

- F_CompSemiTrail_a_V1_0_1_CompSimpackSemitrailingVehicle.c (variant Semitrailing Arm model)
- F_CompMacroJoint_a_V1_0_1_CompSimpackMacroJointVehicle.c (variant Macro Joint Suspension model)
- F_CompLookUp_a_V1_2_0_CompSimpackLookUpVehicle.c (variant Look-Up Table model)

The C code files of all variants may be found in the LCO project folder:

- <Project Folder>\Target_RTPC\SimulinkModels\Simulink_vehiclemodel\.

Please have a look to the following code example to see what change need to be made to let have the build operation finishing successfully:

```
static void mdlInitializeSampleTimes(SimStruct *S)
{
    //real T *h;          /* sampling rate (step-size)          */
    real T h;           /* sampling rate (step-size)          */

    if ( sim_typ == 2 ) /* S-fcn passes equation of motion, which will be integrated by SIMULINK */
    {
        /* In case of continuous S-function block */
        /* ----- */
        ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME); /* pass to Matlab: continuous S-fcn */
        ssSetOffsetTime(S, 0, 0.0); /* ... */
    }
    else /* S-fcn passes ohny the y-vector (S-fcn state integration done internally)*/
    {
        /* In case of discrete S-function block */
        /* ----- */
        //h = (SFunctionDT); /* get sampling rate (step-size) from S-function dialogue par. */
        h = (SFunctionDT); /* get sampling rate (step-size) from S-function dialogue par. */
        ssSetSampleTime(S, 0, h ); /* pass to Matlab: sample time */
        ssSetOffsetTime(S, 0, 0.0 ); /* pass to Matlab: */
    }
}
```

Open the affected file with an appropriate editor. Change the 2 lines as shown in the picture:

- a) Set a comment character at the beginning if the line.
- b) Add the new definition/function access as shown
- c) Store the file and select Build LCO PROJECT in LABCAR-OPERATOR again.

ASCET Module: Changed path for labels of exported (project global) parameters in ASCET Modules using ASCET >V6.0.1HF13

Global parameters may be used to share parameters (e.g. T_VehicleSpeed__r_Wheel) between several model blocks. In ASCET-MD® this may be done by setting the property "Scope" of one parameter to "exported" while all other occurrences of the parameter in other components of the project have this property set to "imported". With the export of a parameter it is visible in the project global context no matter in which of the components – in which level of hierarchy - it has been originally defined.

Starting with ASCET V6.1.0 such parameters are presented in the Experiment Environment's ASCET Module folder always on top level, while in previous versions these parameters were placed in the folder of the ASCET component the parameter have been exported. When updating ASCET from V6.0.1 to a higher version you need to adapt all references to global parameters in the mapping files and instruments. If you use scripts or access global parameters via automation client you need to adapt the respective documents too.

User defined code for disabled CAN messages not migrated

When migrating a LCO project (LCO V5.x) to LABCAR-OPERATOR V5.4.0 user defined code is not included in the transfer in case the check box "Enable/Disable Message for Code Generation" in LCO-IP CAN Editor tab has been set to *unchecked* in LCO V3 or V4 once. This can lead to a problem as soon as the CAN message is enabled for code generation again (and the setting "User Code – Physical/Electrical" is still set to YES). The missing code need to be transferred manually to be prepared for later use.

Standalone migration of previous EE to newer EE version not possible

It is not possible to migrate a LABCAR-EE Workspace from a previous LCO version by just open it in the LABCAR EE installed with LABCAR-OPERATOR V5.4.0. For a full migration the complete LABCAR project need to be opened in LABCAR-IP. After migrating the LCO-IP related documents the executable code for the simulation experiment need to be rebuilt. After successful build a migration of the EE workspace belonging to the project will be triggered automatically.

Migration of Projects with VXI based (ES4350) to VME (ES1651)

In case a LABCAR-OPERATOR Project is migrated from an ES4350 with several Piggy Backs to a VME LABCAR containing two ES1651 PWM boards there might be a wrong order of Processes in the Real-Time OS-Settings, since the necessary dependencies between Carrier boards and piggybacks are sometimes not considered.

To avoid any issues here the user shall reassign all processes:

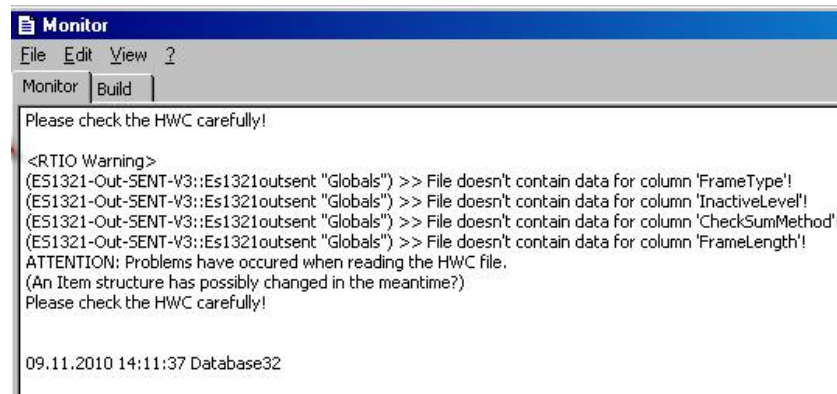
In the OS Settings dialog all processes have to be moved to the left side (Multi-Selection is possible). Afterwards all processes on the left side of the window have to be selected and the "Auto Attach" functionality (context menu) has to be used. All processes are automatically assigned to the respective tasks in a proper order.

New platform Signal Generator does not fully support Time Slices

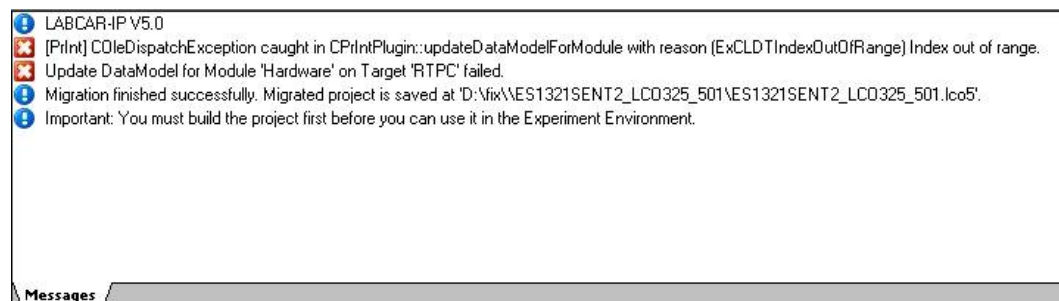
With signal specification of previous Signal Generator (LCO V<5.1.0) (Stimuli Set) it was possible to define Time Slices providing time loops for a group of signals for a specific range of time. Since LABCAR-OPERATOR V5.3.0 it is possible again to specify the loops for the whole duration but not for a specific time range.

ES1321-SENT-RTIO devices need to be replaced manually when using LCO project before V5.0.1

Starting with LCO V5.0.1 the PWM I/O board ES1321 supports the processing of SENT signals according SAE J2716 (JAN2010) standard (also called SENT V3). If you open a hardware configuration (HWC) that contains a RTIO device of the earlier SENT V1 or V2 (in RTIO items tree called "ES1321-In-SENT" and "ES1321-Out-SENT" you will get warnings similar to the following in the RTIO Editor Monitor window:



When migrating a project created with a version of LCO before V5.x.y you will get an additional error during migration and build in the LCO-IP view "Messages":



It is required to remove the "ES1321-In-SENT" and/or "ES1321-Out-SENT" and replace them by the new "ES1321-In-SENT-V3" and "ES1321-Out-SENT-V3" items.

FlexRay Module: New option "Generate Target Communication flags as variables" may require adaptations in Experiment Layers after a LCO project is migrated from earlier LCO versions

With LABCAR-OPERATOR V5.1.2 the user can decide whether control signals should be presented in EE as Parameter elements or LABCAR inports. Nevertheless this setting influences all the control signals uniform. With the default setting "Generate Target Communication flags as variables" = INACTIVE LABCAR inports are created for the "FrameEnable.." Signals and the NIF control signals in folder "Target". If the setting is switched to ACTIVE Parameters are used as element type instead.

This is different to the situation in previous versions where the LABCAR-NIF control signals in folder "Target" where always Parameters and the "...Enable..." switches where always LABCAR inports. If you have control signals assigned to instruments in the Experiment you need to change the assignment according to the new element type.

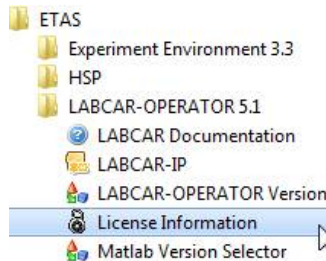
4.4 Discontinued Features

Support of external tools

For version of supported external tools please have a look to the most current version of "LABCAR-OPERATOR V5 - Software Compatibility List".

MS Windows™ start menu entry “License Information” removed

Starting with LABCAR-OPERATOR V5.2.0 the start menu item “License Information” is removed:



You can obtain the license information required for registration of your license in the ETAS license portal by starting the License Server from the start menu and selection of menu TOOLS-OPBTAIN LICENSE INFO instead.



Migration support for LCO V3.2.x and V4.x.y projects

In LABCAR-OPERATOR V5.4.0 there is no migration support for LCO projects that have been created or have been process last time with versions before 5.0.0. To use such projects in LABCAR-OPERATOR V5.4.0 you need to migrate to a version LABCAR-OPERATOR V5.0.x or V5.1.x first and update to LABCAR-OPERATOR V5.4.0 afterwards. Starting with LABCAR-OPERATOR V5.2.0 a previous V5.1.x or V5.0.x version can be installed in parallel to LABCAR-OPERATOR V5.4.0.

5 Hints

This section contains special hints regarding the usage of LABCAR-OPERATOR and latest news, which are not described in the manual.

Please have also a look to the examples in the MISC folder on the installation media.

5.1 User Rights to use LABCAR-OPERATOR

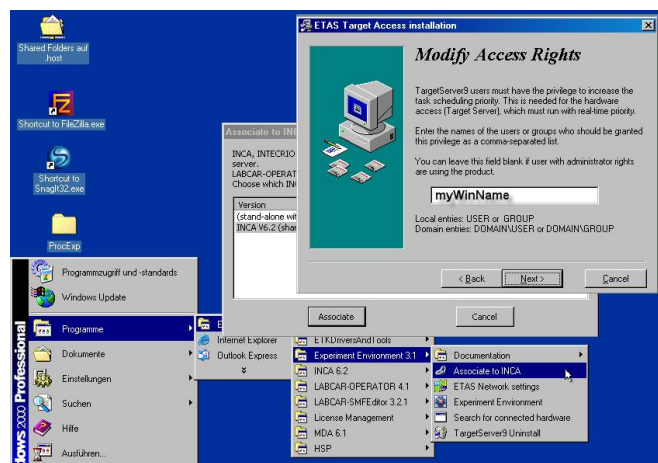
When creating a new user or a new group in the WINDOWS user management certain default user and access rights are granted according to the security policy applied in the WINDOWS operating system. These default rights are depended on the membership of the PC in a work group or a domain and may even vary depending on the certain WINDOWS version used. In the past we experienced the following additional user and access rights elevations need to be applied:

For execution of simulation experiments using LABCAR-EE

- As minimum the set of user rights granted for a standard user account need to be granted (see WINDOWS 7 example in screenshot):



- The user right "Increase Scheduling Priority". This enables the real-time measurement access (done by the Target Server) from the host PC with highest possible priority. This can be done immediately after software installation when selecting the appropriate Target Server version in the window "Associate to INCA".



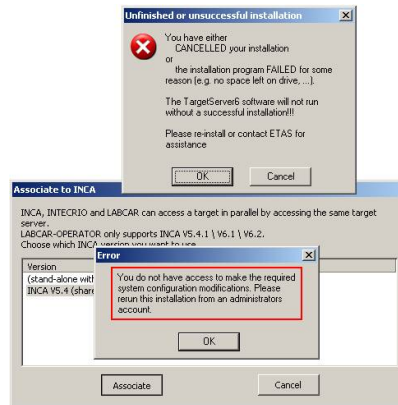
Note: This step is only required if the Experiment Environment is started by a user having no Administrator rights and only for association with INCA versions lower than V7.1.

- The administrator of the computer must select a temporary file folder on which all users who want to work with LABCAR-OPERATOR on this computer have read and write access (e.g. C:\TEMP)
- All users who want to work with LABCAR-OPERATOR on this computer need full access to the LABCAR-OPERATOR project directory and the ETAS program directories (e.g. C:\ETAS, C:\Program Files\ETAS).

For changing installation settings of LABCAR-OPERATOR

- For the operation “Associate to INCA” and “LABCAR-OPERATOR Version Selector” you need administrator rights. Otherwise the following issues occur:

1. Associate to INCA: After activation of button ASSOCIATE the following error message are displayed:



2. LABCAR-OPERATOR Version Selector: In the application window no software version is displayed:



Important Note:

You may perform the activities “Associate to INCA”, “Switch LCO version” and “Matlab Version Selector” even when logged in with standard user rights by starting the affected applications with “Run as Administrator” (Windows 7). In this case cycling a log-off / log-in or restart of Windows OS is required to apply the changes for all of the affected user groups.

5.2 Issues with LABCAR-CCI (Calibration Connector for INCA)

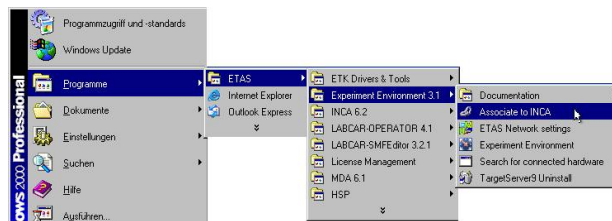
- During working with both LABCAR-OPERATOR and INCA at the same time, there are certain operations, e.g. working page operations, which have to be performed in INCA as they are not yet supported in the EE of LABCAR-OPERATOR. However, please restrain from performing experiments in INCA while experimenting with LABCAR-OPERATOR as unwanted side effects will be possible.
- If there are two INCA versions installed, LABCAR-OPERATOR will automatically open the latest installed INCA version. LABCAR-OPERATOR V5.4.0 comes with an “Associate to INCA” functionality (to be found in the Start Menu via Programs/ETAS Experiment Environment 3.x/Associate to INCA) to switch between different INCA versions.
- If an INCA project is opened with a newer INCA version (e.g. 7), it is not possible to open it again with a former INCA version (e.g. 6).
- When modifying the hardware used in an INCA experiment, it is necessary to import the changed INCA experiment again to get an update on the valid and available INCA measurement and parameter labels.
- The recording will not be processed correctly if the stop trigger condition is fulfilled a short time (~1s) after the start trigger condition. This may be the situation if you use a LABCAR signal for

the start-trigger and an INCA signal to generate the stop trigger. In this case there may be insufficient data for offset correction (EHI # 254170).

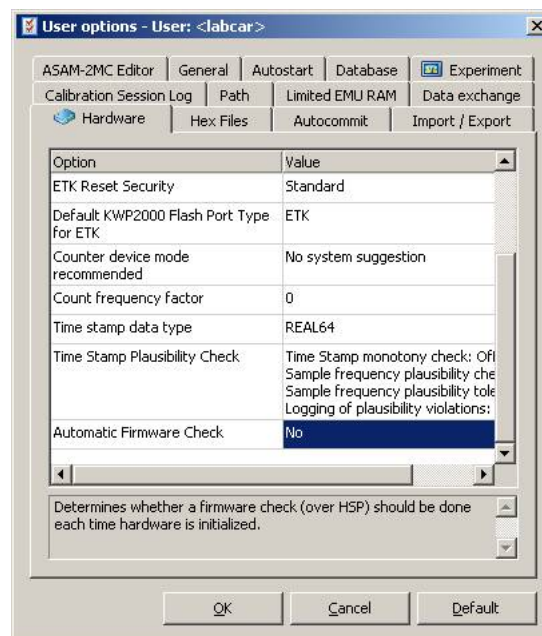
- Avoid blank spaces in INCA hardware device names (HWK). If a device name contains a blank space signals generated by this device cannot be used as start or stop trigger in Experiment Environment Datalogger.
- Reading and writing of string based data types in INCA are not supported yet.
- It is not supported to manually set the value of an enumeration parameter in the EE e.g. via table instrument (EHI # 328285).

5.3 Issues operating LABCAR-OPERATOR and INCA in parallel

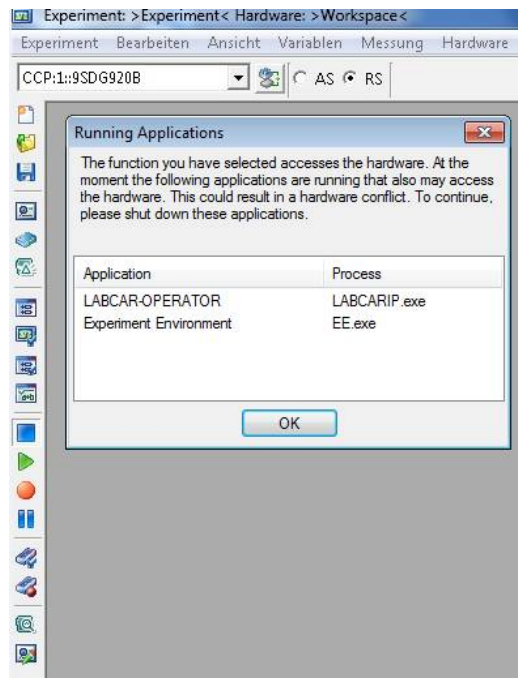
When working with both LABCAR-OPERATOR and INCA at the same time, both tools need perform the processing of measured data with a TargetServer tool which is compatible to both applications. So even if you do not want to have INCA data in the LABCAR-EE you must synchronize the TargetServer package installation by executing the operation "Associate to INCA".



INCA >= V7.0.0: To avoid blocking automated operation due to a confirmation dialog please make sure that the Automatic Firmware Check is disabled when running EE and INCA in parallel.



Otherwise a warning dialog will be displayed when the firmware check is activated.



Anyway you may just confirm the warning dialog to continue working.

• **Automatic Firmware Check**

Determines whether INCA shall check the firmware version each time a new hardware is initialized during the current INCA session.

Yes	When the hardware gets initialized (e.g. upon opening an experiment), INCA checks automatically the firmware status of the corresponding hardware and checks if firmware version and INCA version are compatible. The overall result of the firmware check will be indicated in the lower icon bar of the INCA Experiment Environment. Automatic firmware checks require HSP V9.0 or higher.
No	INCA does not perform an automatic firmware check. You can check and update the firmware status manually from the Hardware Configuration Editor or the Experiment Environment.

5.4 Error message “libut.dll not found” after installation of MATLAB®/Simulink®

To avoid this error message on start of a MATLAB®/ Simulink® command window (while performing BUILD LCO PROJECT or EDIT PROJECT even if the installation path was added to the OS environment variables) after installation of MATLAB®/ Simulink® at least LOG OFF/LOG ON in WINDOWS is necessary.

5.5 Multiple Versions of The MathWorks Releases on one PC

LABCAR-OPERATOR allows cooperating with different versions of MATLAB®/Simulink®. If several versions of MATLAB®/Simulink® are installed, different LABCAR-OPERATOR projects may use them independently of each other. However, it is not supported to have Simulink® Modules of different Simulink® versions within one LCO project. With the new MATLAB® Version Selector you can decide easily which of the supported versions MATLAB®/Simulink® should be used for working with LABCAR-OPERATOR.

5.6 Use of LABCAR- In-/Output Blocks for Simulink® models without having LABCAR-OPERATOR involved

The LABCAR-In-/Output Blocks will be available in the Simulink® Block library per default only, if you open a SIMULINK® model by "Edit Model" operation in LABCAR-OPERATOR. If you want to open a Simulink® model without LABCAR-OPERATOR, you should add the folder "C:\Program Files\ETAS\LABCAR-OPERATOR5.4\LABCAR-IP\SiCo" (or the equivalent if you did not install LCO with the default settings) in the Matlab® search path, to have no error message when opening a model with LABCAR- In-/Output Blocks and to have the blocks included into the Simulink® Library Brower.

You may add the path either with the menu item ADD PATH (see Simulink® Help for more information) or by entering the instruction

```
>> addpath(' C:\Program Files\ETAS\LABCAR-OPERATOR5.4\LABCAR-IP\SiCo');
```

in the Matlab® command line. When opening the software Matlab/Simulink® all paths in the search path are scanned for files with the name "sblocks.m". These files contain an incremental description of user defined blocks what may be integrated into the Simulink® Library Brower.

5.7 How to remove missing Ethernet board from WINDOWS Device Manager

If the Ethernet board used for connecting ETAS hardware to PC have been removed from PC without formal de-installation it is not possible to re-use the fix IP address for a new board because the previous board is not displayed in WINDOWS Device Manager anymore, nevertheless the software device for the board is still present and the IP address 192.168.40.240 remains assigned to the board. To remove the redundant software device, the following procedure may be started:

1. Create a batch file, e.g. "devmgr.bat"

2. Insert these two lines:

```
"set devmgr_show_nonpresent_devices=1"  
"devgmt.msc"
```

3, Execute "devmgr.bat".

4. Select in WINDOWS Device Manager menu item VIEW/ANSICHT-DISPLAY HIDDEN DEVICES/AUSGEBLENDEDE GERÄTE ANZEIGEN

Now they can be safely removed and the IP address is released to be used for other devices.

5.8 RTPC simulation target: Rules for OS setting in LCO

For the simulation target RTPC the ERCOS specific scheduling scheme is ported to the scheduling scheme of Real-Time LINUX. Due to architectural differences of both OS there is no real 1:1 transformation of the task and their properties into the Linux world. In addition to that the current connection between RTPC and VME I/O hardware requires a more serialized signal transfer between to RTPC and VME hardware (ES1130 via Ethernet) to have the optimum performance regarding the hardware latency time. The following rules should be obeyed to have the optimal runtime behavior:

- All hardware processes should be assigned to that task which is selected as "ES1130 SYNCHRON TASK" for the ES1130 sub system in the RTIO Editor. In particular the hardware control signals, which ought to be assigned to a special control task with higher period dt, should preferable be assigned to the "ES1130 SYNCHRON TASK" if RTPC is used as simulation target.
- All model processes (time periodic processes of SIMULINK or C Code modules) which have a direct connection to a HWC port should be assigned to the "ES1130 SYNCHRON TASK". Please take into consideration that the task used for these modules must be

selected in the source code editor, either the SIMULATION OPTIONS dialogue in SIMULINK or in the C Code Module Interface Editor.

- The “ES1130 SYNCHRON TASK” should have the highest priority and a preemptive scheduling mode.
- For LABCAR-NIC (the module “CAN” in the LCO Project explorer) the processes should be assigned to a task of their own. This is because the only hardware supported by LABCAR-NIC today is the PCI based IXXAT CAN board. Nevertheless it may be also assigned to the “ES1130 SYNCHRON TASK” if the LABCAR-NIC base rate is equal to the period of this task.

5.9 Problems when assembling Ethernet components with different transmission rates: Operation “Save parameters” doesn't work

In a hardware setup components linked with different transmission rate, e.g. RTPC (1Gbit) -> Switch (1Gbit) -> PC (100Mbit) problems may occur while saving large parameter arrays. Instead a timeout exception is thrown. For instance packages arrive with 1Gbit at the switch, need to be forwarded to the PC with only 100 Mbit, some packages may get lost, when processing values of e.g. a big curve (16kb of data). In that case the UDP message will be spitted into n parts on Ethernet. Due to the limited buffer size packages may get lost without being able to restore the UDP message content.

5.10 Different subscription of array indices in LABCAR-OPERATOR Experiment Environment and LABCAR-PARAMETRIZATION ASSISTANT

The array index count in LABCAR-OPERATOR Experimental Environment and LABCAR-PARAMETRIZATION ASSISTANT is different: While the count of elements for arrays in Experimental Environments widget Table Editor starts with 0 in LABCAR-PARAMETRIZATION ASSISTANT it begins with 1 instead. This should be taken into consideration when comparing arrays or tables in both instruments.

5.11 Issue with borrowing of LCO concurrent license

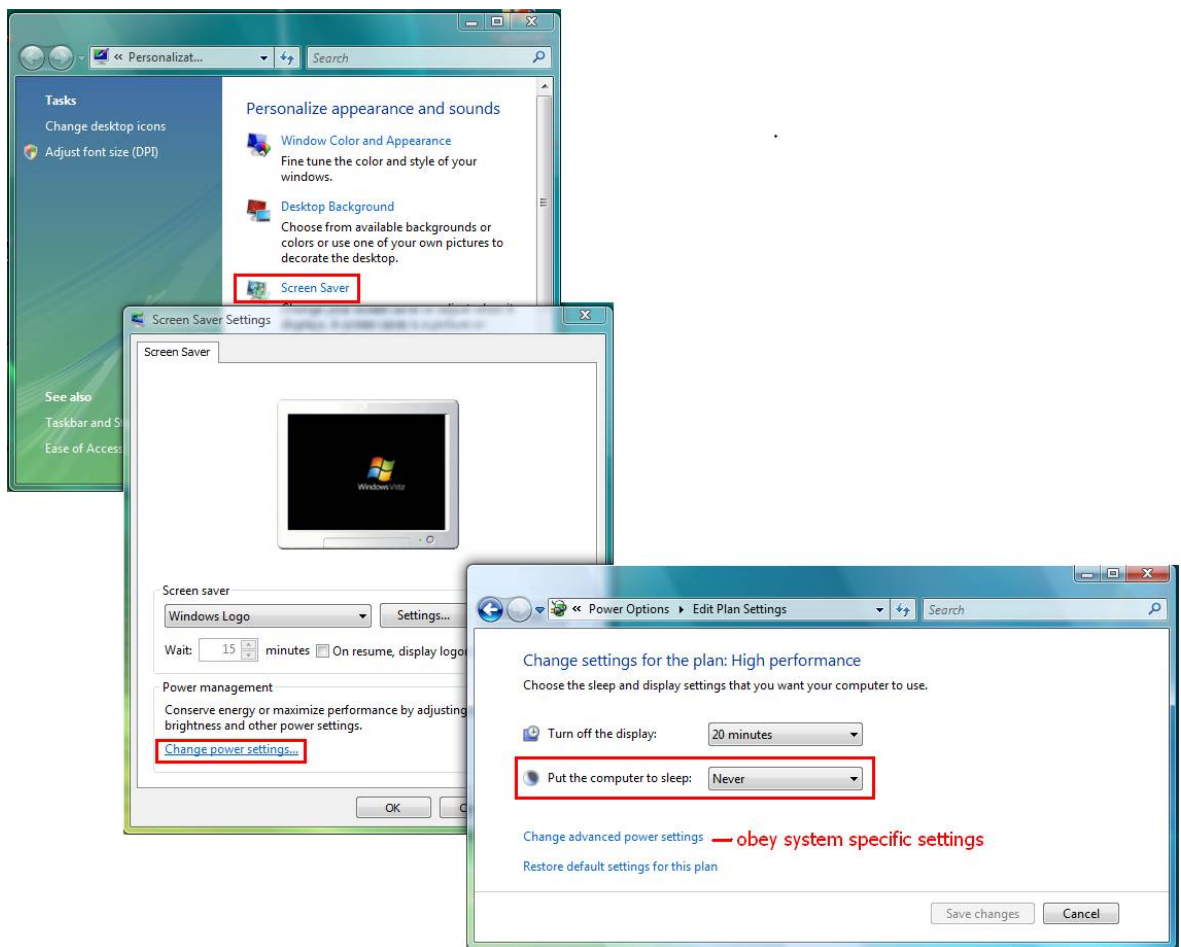
LABCAR-OPERATOR may be used with the concurrent perpetual software license in case the PC with the installation of LCO is connected to a network running a license server and there are still licenses available. To continue working with LCO even if the PC got disconnected it is possible to check out a license from the license server for a certain time. This is called “borrow a license”.

Please be aware, that this borrowed license grants the license only for the WINDOWS user who checked out the license. It is not possible to check out a license and having an alternate use of the borrowed license by different users of the PC. This is also true for situations where LCO is started with another account for instance the user “system” or another accounts. If any part of the software is started via script it needs to be run with the same user account the license was borrowed before.

5.12 Deactivation of PC Power Management for unattended operation

To avoid unintended interruption of connection between simulation target (RTPC) and LCO PC during an experiment session and ensure stable operation, we recommend deactivation of all settings which limit the functionality or turn of hardware devices like energy saving options or power management control. There are different places where these options may be controlled in general or specific for a certain device.

This recommendation does not include the screen saver functionality which locks the console to allow unattended operation during automated test.



Depending on the specific hardware the connection will be interrupted for sure if the OS system is turned in the states "standby" and "hibernation".

5.13 Open a project without a license: IP is closed without information.

If a LCO project is opened by double click the project file in MS Windows Explorer and there is no valid license for LCO-IP available, LCO- IP is opened for short and closed again without notice to the user. This is the expected behavior for automated operation of LCO-IP. Nevertheless a warning may be found in the log file of LCO-IP.

5.14 FlexRay™ Residual Bus simulation with EB tresos® Busmirror® containing instances of signals

When assigning a signal to different Signal Groups/PDUs/Frames in the corresponding editors of EB tresos® Busmirror® the signals are not instantiated automatically. Instead it is assumed the the very same signal (instance) transporting identical data to the different frames shall be distributed.

If the user want to use a signal as a template he need to create a copy of the signal for each signal instance and assign the individual copies to the different Signal Groups/PDUs/Frames. That way individual data for each copy can be transported.

Nevertheless in LABCAR-EE both use cases are treated in LABCAR-OPERATOR in the same manner: Signals are represented as LABCAR ports in the folders created for frames in LABCAR-IP Connection Manager and LABCAR-EE Workspace Elements for each frame. Even if it is the very same signal it

may be found in every frame folder the signal is assigned. This is to pursue the same systematic relation frame-signal as during creation of communication matrix in the FlexRay™ authoring tool and to make it easier to find the signal in the Workspace Elements tree.

When configuring sent frames (Tx) containing a the identical signal this leads to the paradox situation that more than one LABCAR Inport is created for each frame. To ensure correct sequence of data processing the following remarks should be taken into consideration:

1st use case: All Inports shall be connected to the same source Outputport:

ALL Inports need to be connected to the driving source Outputport, you must not connect just one LABCAR Inport, the option "Optimize Code generation for signal value comparison"=" Switch transformation mode for Sent signals to spontaneous" does not care.

OR

Connect just one of the Inports to the driving source Outputport but set the option "Optimize Code generation for signal value comparison"=" Switch transformation mode for Sent signals to spontaneous" to TRUE.

1. 2nd use case: The inports shall be connected to different source Outputports (merge function)

Connect one, more or all Inports to all source outputports. You must set "Optimize Code generation for signal value comparison"= " Switch transformation mode for Sent signals to spontaneous" to TRUE. This ensures that the value refreshed at last will be transmitted.

With EB tresos® Busmirror® you can even have different signal instances among different PDUs. To assign them correctly to LABCAR ports created for these instances you need to introduce user defined code to process the signal instances correctly. See the following code fragments addressing two signal instances intended to process the receive and the send side of one and the same signal:

```

//[declarations]
int handleSendSignal      = 0;
int handleReceiveSignal   = 0;

//[before:SendSignalGroup:<Name of send SignalGroup or PDU>]
handleSendSignal = 1;

//[after:SendSignalGroup:<Name of send SignalGroup or PDU>]
handleSendSignal = 0;

//[before:ReceiveSignalGroup:<Name of receive SignalGroup or PDU>]
handleReceiveSignal = 1;

//[after:ReceiveSignalGroup:<Name of receive SignalGroup or PDU>]
handleReceiveSignal = 0;

//[before:SendSignal:<Name of send SignalInstance>]
if(handleSendSignal) {
    // handle signal
}

//[beforeEncode:SendSignal:<Name of send SignalInstance>]
if(handleSendSignal) {
    // handle signal
}

//[beforeSend:SendSignal:<Name of send SignalInstance>]
if(handleSendSignal) {
    // handle signal
}

//[after:SendSignal:<Name of send SignalInstance>]
if(handleReceiveSignal) {
    // handle signal
}

//[before:ReceiveSignal:<Name of receive SignalInstance>]
if(handleReceiveSignal) {
    // handle signal
}

```

```
//[afterReceive:ReceiveSignal:<Name of receive SignalInstance>]
if(handleReceiveSignal) {
    // handle signal
}

//[afterDecode:ReceiveSignal:<Name of receive SignalInstance>]
if(handleReceiveSignal) {
    // handle signal
}

//[after:ReceiveSignal:<Name of receive SignalInstance>]
if(handleReceiveSignal) {
    // handle signal
}
```

5.15 Change of FlexRay™ Tool Chain Version triggered only for supported versions

According to the setting "Tool Chain Version" in the FlexRay™ Module Editor the TAL version is switched to the TAL version that corresponds to the EB tresos® Busmirror® version the FlexRay™ project have been created. Nevertheless in cases where the tool chain version cannot be determined (either there is a version used not supported by LCO-NIF or there is a technical problem) the tool version is changed neither in the FlexRay™ Module Editor nor on the RTPC before built. You need to set the correct TAL version on RTPC in the RTPC Web Interface in this case. You can perform this either interactively, opening a special RTPC Web Interface by clicking on the following link in the web browser of RTPC:

```
http://192.168.40.14/cgi-bin/eb-tal-select?format=html
```

or non-interactive by emitting a http-request in Web Browser, example for TAL version corresponding to EB tresos® Busmirror V4.6.1:

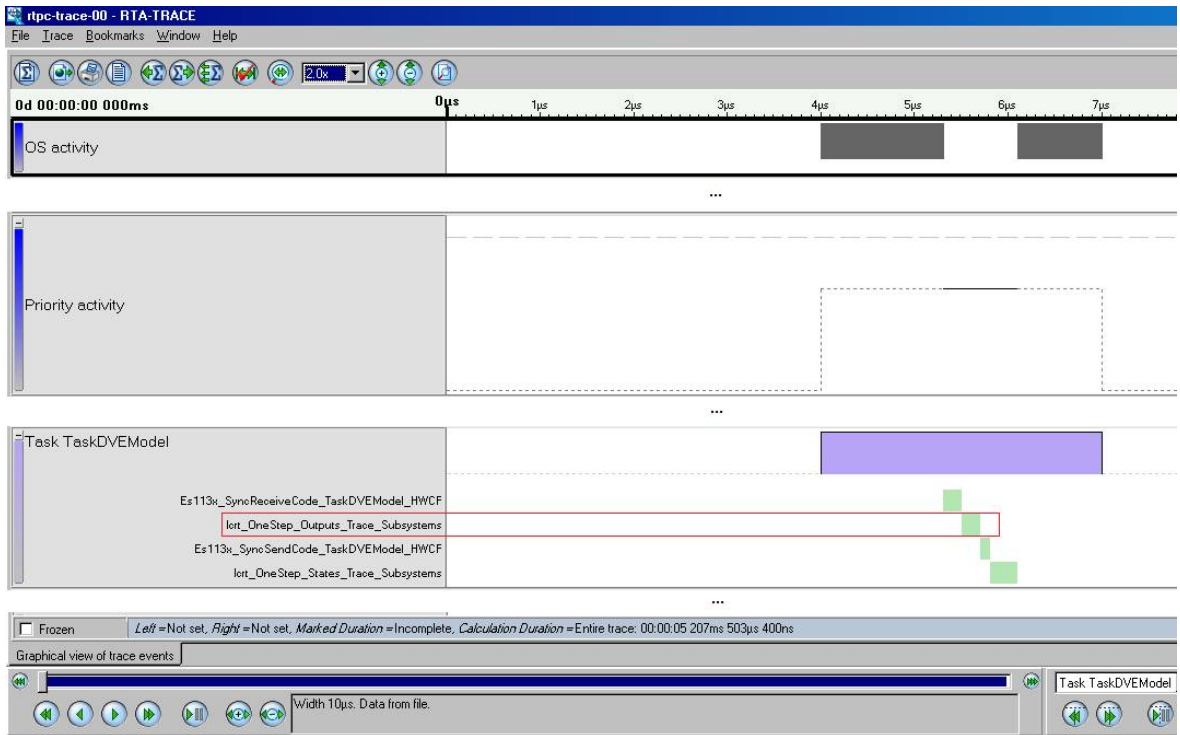
```
http://192.168.40.14/cgi-bin/eb-tal-select?cmd=set&version=4.6.1
```

or EB tresos® Busmirror V4.8.1:

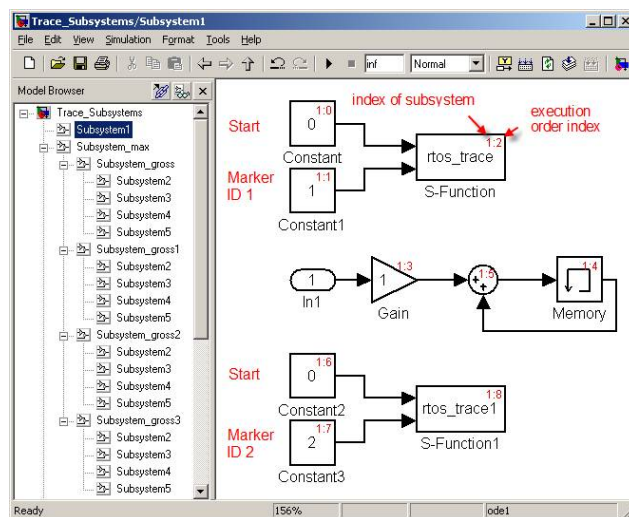
```
http://192.168.40.14/cgi-bin/eb-tal-select?cmd=set&version=4.8.1
```

5.16 Profiling of Simulink® Subsystems (EHI # 314964)

With the Runtime Trace functionality of RTPC it is possible to investigate the runtime of tasks and processes scheduled in the OS tab of LCO-IP for a specific project, see the screenshot taken from ETAS RTA-Trace below:



The process runtime for “lctrl_OneStep_Outputs_Trace_Subsystems” indicates the runtime of code processed in the global context of the Simulink® models callback method “mdlOutputs”. To resolve execution times in more details you can add calls to the trace C API within your model to set marker in the RTPCs Runtime Trace log output. Look at the following Simulink® model example:



C code S-Functions may be used to call the RTPC trace C API methods and set markers with the ID with the following prototypes:

```
rtos_trace_interval_begin(int id, int arg);
rtos_trace_interval_end(int id);
```

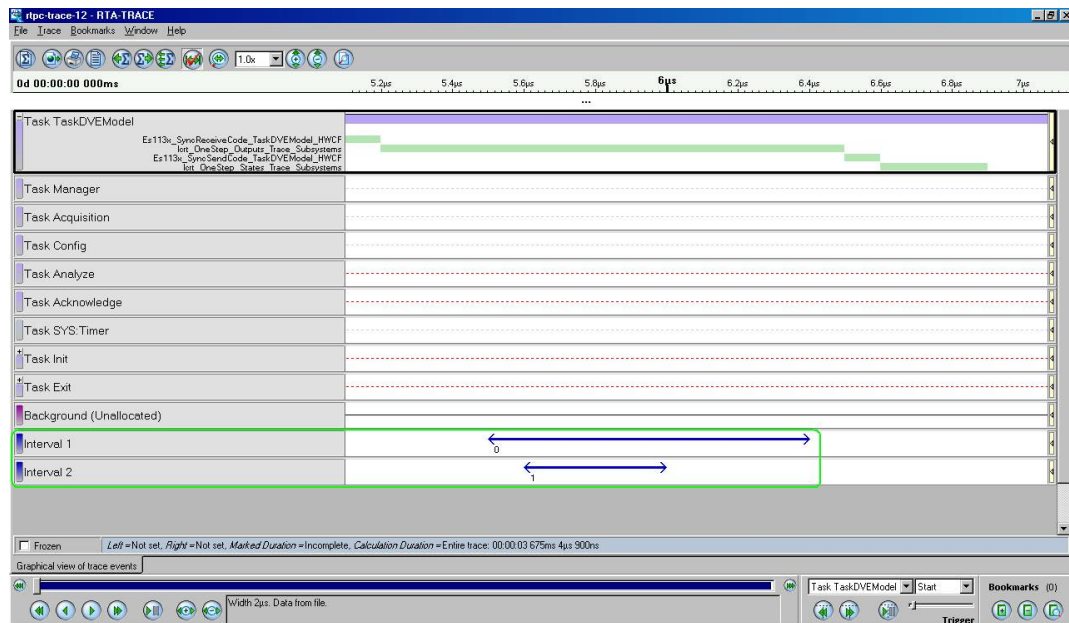
This places the interval begin/end points with the marker ID in the RTA trace file. In the example the S-Functions “rtos_trace” and “rtos_trace1” are controlled by its inputs: First Input is used for indicating the begin (=0) or end (=1) tracing; the second input is used for choosing the ID (int). See

the following code lines as C code S-Function implementation example:

```
static void mdlOutputs(SimStruct *S, int_T tid)
{
    InputRealPtrsType uPtrs0 = ssGetInputPortRealSignalPtrs(S,0);
    InputRealPtrsType uPtrs1 = ssGetInputPortRealSignalPtrs(S,1);

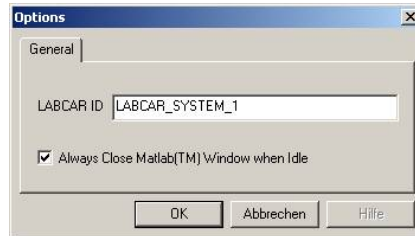
#if !defined(MATLAB_MEX_FILE)
    int ID;
    ID=*uPtrs1[0];
    if (0==( *uPtrs0[0])) {
        rtos_trace_interval_begin(ID, 1);
    }
    if (1==( *uPtrs0[0])) {
        rtos_trace_interval_end(ID);
    }
#endif
}
#endif
}
```

Running the code on RTPC the following RTA trace log output is produced indicating the interval between *start* and *stop* for the two intervals (screenshot from ETAS RTA-Trace):



5.17 Avoid Tesis model license error

To avoid multiple checkout of licenses when working with simulation models of the Tesis tool family the option "Always close Matlab™ Window when Idle" should be set active.



You can set this option in the "Options" window opened with the menu item TOOLS-OPTIONS in LABCAR-IP.

5.18 Limits for curves and maps in Experiment Environment

Depending on the tool used for creation of a LCO Behavior Module there are different limits for the maximum size of each dimension of a characteristic curve or map. Nevertheless there is also a limit due to the communication bandwidth when transferring values of a curve/ map between RTPC and LCO-PC. For maps the maximum size is 7950 sampling points, for curves it is 89*89.

Parameter Elements that exceed this limit can neither displayed in EE instrument nor processed via parameter download. You can find corresponding error messages in the EE log files or Application Log view. Please consider that log messages may be issued after the download so that you need to scroll in the log window view to find the error messages regarding the parameter download.

5.19 Issues with LABCAR-LCE/LCX (LABCAR Integration ETK/XETK)

- Preparation of ECU project for LCO-LCX/LCE with EHOOKS
To enable stimulation of ECU quantities bypass code need to be inserted into the ECU software. For use with LABCAR-OPERATOR we recommend to select "External Bypass" as Hook Type. If a parallel ETK is used as ECU interface one can also select "Parameter".
- Preparation of ETK/XETK
Make sure that the ETK is running on the working page by default. FiL can only calibrate when it is on the working page. This can be ensured by using the ETK config tool. That configuration need to be saved the ETK to be available directly after powering on the ETK.

5.20 Issue on play back of scripts with Experiment Environment Script Recorder

The following issue have been observed with a certain PC: When try to play back the script with PLAY SCRIPT an error message is issued: "Error in ScriptRecorder state machine PlaybackScript : Cannot start process because a file name has not been provided", the script is not started. The reason for this issue may be that under some circumstances (like installing updates and / or service packs) the installation of the Windows Scripting Host may get corrupted. The following steps way be used to fix the problem.

!! Be warned that this will change the registry of your machine. You may create a backup of the complete registry before you do the following steps. You also may create the keys described below

manually !!

1. Create a text file with these contents:

```
Windows Registry Editor Version 5.00

[HKEY_CLASSES_ROOT\.vbs]
@="VBSFile"

[HKEY_CLASSES_ROOT\VBSFile]

[HKEY_CLASSES_ROOT\VBSFile\DefaultIcon]
@="%SystemRoot%\System32\WScript.exe,2"

[HKEY_CLASSES_ROOT\VBSFile\ScriptEngine]
@="VBScript.Encode"

[HKEY_CLASSES_ROOT\VBSFile\Shell]

[HKEY_CLASSES_ROOT\VBSFile\Shell\Open]

[HKEY_CLASSES_ROOT\VBSFile\Shell\Open\Command]
@="%SystemRoot%\System32\WScript.exe" "%1" %*

[HKEY_CLASSES_ROOT\VBSFile\Shell\Open2]

[HKEY_CLASSES_ROOT\VBSFile\Shell\Open2\Command]
@="%SystemRoot%\System32\CScript.exe "%1" %* "
```

2. Name the text file "VBSFile.reg".
3. Execute the file by double click on the file in Windows Explorer

The file will add the registry keys needed to retrieve the path to the Windows Scripting Host for .vbs files.

5.21 "Global Parameters" in LABCAR-MCS based LCO projects (Globally shared workspace names)

Global parameters are used to share parameters (e.g. T_VehicleSpeed__r_Wheel) between several model blocks. In Simulink® the parameters are shared by use of global workspace parameters.

If you want to share parameters in a RTW (Real-Time Workshop) exported Simulink® model, the *Inline* option has to be activated in the Simulink® model and all parameters have to be defined as tunable parameters in the selection list.

This procedure, first introduced in ETAS Simulation Model DEVM V5.0.1, has to be adapted for other ETAS models GEVM V5.0.2 and VDYM V5.0.1:

1. When you open the Simulink® model "vehiclemodel.mdl", in the model preload callback function all workspace model parameters are loaded by processing

respective m-files (the m-file may be found in the folder "Misc/Exported Parameters Handling" on the model product CD).

2. After loading all workspace model parameters, they are defined as tunable see the following code lines:

```
...  
CreateInlineParameterFile(whos('*_*_*'), 'InlineParameters.txt');  
AddInlinedParameters('InlineParameters.txt');  
...
```

This may be done in the model post-load call back function.

After the integration of the Simulink® model into LABCAR-OPERATOR, all parameters can be selected in the LCO-EE Workspace Elements tree.

NOTE: *Inlined* parameters which are used at least twice in the model will be collected in the sub-folder "_GlobalParameters" in the LCO-EE Workspace Elements tree.

5.22 How to update link for Simulink Module imported as reference

Starting with LCO V5.1.2 a Simulink model can be imported into an LCO project as link (checkbox "Copy model directory into project" in the "Add new Module Wizard" is inactive). When the LCO project is relocated to another PC the reference might need to be adapted. Simply open the file "[project name]\Target_RTTPC\SimulinkModels\Simulink_[Simulink model name]\[Simulink model name].conf" and change the link:

```
[model]  
model.dir=d:\projects\mySimulinkProjects\simulink_model_1\  

```

in a way it points to the path where the model to be used is situated.

5.23 How to change reference behavior of Simulink Model imported as link

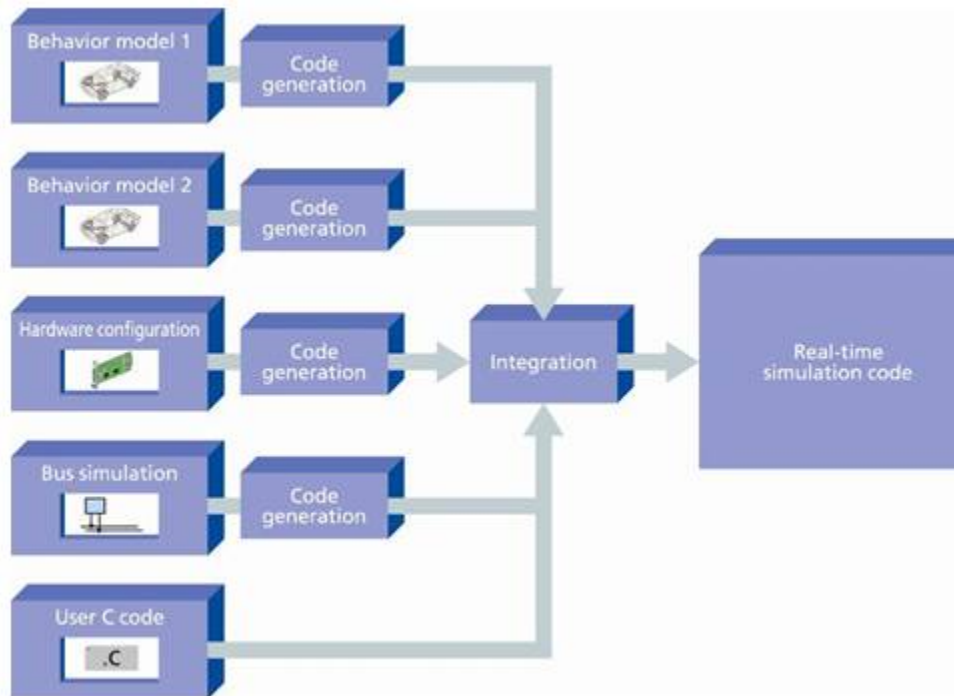
Starting with LCO V5.1.2 a Simulink model can be imported into an LCO project as link (checkbox "Copy model directory into project" in the "Add new Module Wizard" is inactive). If you decide to prefer having a copy instead of a link at a later time you need to re-import the Simulink model.

To avoid losing connections proceed in the following way:

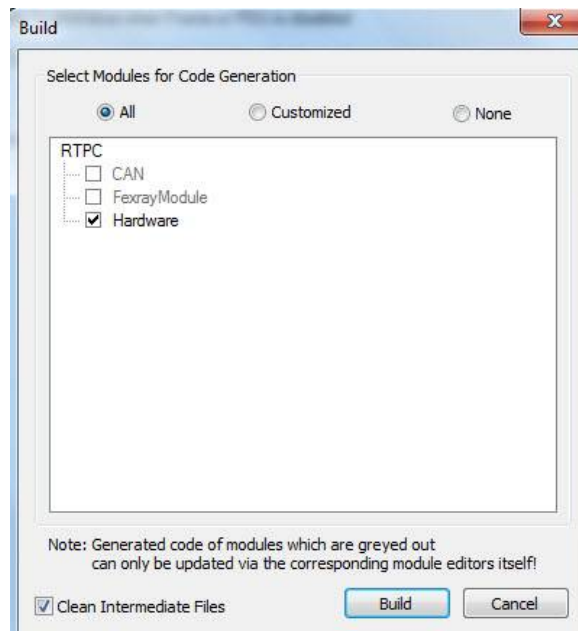
1. Export existing connections in Connection Manager (check all, right mouse click-export)
2. Erase the Simulink module
3. Add Simulink Module with option „Copy into project..." active
4. Import connections again

5.24 Meaning of checkbox settings for code generation

In the code generation dialog the modules may selected new C code is to be generated before starting the build process.



You will find the checkboxes for several Module like CAN/LIN or C Code Module disabled. This is because the code generation process is triggered by previous activities (e.g. when saving the module) or only the interfaces in global project environment- are to be generated (C-Code module). The checkbox state therefore reflect the situation that the user defined C code for the modules is taken as is and does not need code generation as shown in the following picture:



5.25 How to exclude files to be copied to codegen directory and RTPC

A template for a blacklist file with name "exclude.txt" is provided in folder (with default installation settings):

C:\Program Files\ETAS\LABCAR-OPERATOR5.2\BuildSystemTools\make\templates

containing wildcard capable patterns to exclude files to be copied during the build process. Please notice: The file is only considered if it has been placed into the folder "Target_RTPC" of a certain LCO project.

Please see the content of the file (LCO V5.2.0) as an example:

```
#####  
#  
# Exclude.txt  
#  
# This file contains a list of exclude patterns. Files  
# matching one of these patterns will be excluded from  
# file transfer between Host-PC and target RTPC.  
#  
#####  
  
*.bak  
*.log  
*.m  
*.t1c  
.svn/
```

You may enhance or reduce the file by including an new line/erasing a line with the file name pattern.

6 Glossary and References

6.1 Definitions and Abbreviations

Target

The hardware a program or an experiment runs on, e.g. LABCAR-RTPC

LABCAR-MCS

Product Add-On Modeling Connector for Simulink®, which directly connects MATLAB®/Simulink® to LABCAR.

LABCAR-MCA

Product Add-On "Modeling Connector for ASCET" enables integration of ASCET projects as ASCET Modules in LCO and build of LCO projects with ASCET modules.

LABCAR-RTC

Product Add-On "Real-Time Execution Connector" required for configuration and operation of all real-time applications with LABCAR

LABCAR-CCI

Product Add-On "Calibration Connector for INCA" enabling synchronous M&C and simulation data recording by controlling INCA from LABCAR

LABCAR-NIC

Network Integration CAN, facilitates comfortable configuration and operation of tests for ECU functions requiring CAN Bus communication

LABCAR-NIL

Network Integration LIN, facilitates comfortable configuration and operation of tests for ECU functions requiring LIN Bus communication

LABCAR-NIF

Network Integration FlexRay™, facilitates comfortable configuration and operation of tests for ECU functions requiring FlexRay™ Bus communication

LABCAR-LCX

Product Add-On "LABCAR Integration XCP" enables set up and code generation of XCP Module which is designed for interfacing an ECU with XETK connected via Ethernet.

LABCAR-IP

Abbreviation stands for LABCAR Integration Platform, one of the LABCAR-OPERATORS program windows.

LABCAR-EE

Abbreviation stands for LABCAR Experiment Environment, one of the LABCAR-OPERATORS program windows.

ECU

Abbreviation stands for Electronic Control Unit.

HSP

Hardware Service Pack

EHI

"ETAS Helpdesk International", the ETAS defect tracking tool.

Code Generation

Code generation is the first step in the transformation of a physical model to executable code. The physical model is transformed into ANSI C code. Since the C code is compiler (and therefore target) dependent, different code for each target is produced.

7 **ETAS Contact Addresses**

ETAS Headquarter

ETAS GmbH
Borsigstr. 14
70469 Stuttgart
Germany

Phone: +49 711 3423-0
Fax: +49 711 3423-2106
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
ETAS technical support

WWW: www.etas.com/en/contact.php
WWW: www.etas.com/en/hotlines.php