

dSPACE Release

# New Features and Migration

Release 2017-B – November 2017

## How to Contact dSPACE

Mail: dSPACE GmbH  
Rathenaustraße 26  
33102 Paderborn  
Germany

Tel.: +49 5251 1638-0  
Fax: +49 5251 16198-0  
E-mail: [info@dspace.de](mailto:info@dspace.de)  
Web: <http://www.dspace.com>

## How to Contact dSPACE Support

To contact dSPACE if you have problems and questions, fill out the support request form provided on the website at <http://www.dspace.com/go/supportrequest>.

The request form helps the support team handle your difficulties quickly and efficiently.

In urgent cases contact dSPACE via phone: +49 5251 1638-941 (General Technical Support)

## Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit <http://www.dspace.com/go/support> for software updates and patches.

## Important Notice

This document contains proprietary information that is protected by copyright. All rights are reserved. The document may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the document must not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© 2000 - 2017 by:  
dSPACE GmbH  
Rathenaustraße 26  
33102 Paderborn  
Germany

This publication and the contents hereof are subject to change without notice.

CalDesk, ConfigurationDesk, ControlDesk, MicroAutoBox, MicroLabBox, SCALEXIO, SYNECT, SystemDesk, TargetLink and VEOS are registered trademarks of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

# Contents

<b>About This Document</b>	<b>11</b>
Conventions Used in dSPACE User Documentation.....	11
Accessing dSPACE Help and PDF Files.....	12
<b>Overview of dSPACE Release 2017-B</b>	<b>15</b>
General Enhancements and Changes.....	15
New dSPACE Help Available.....	17
Product Version Overview.....	21
New Product Key Features.....	23
<b>New Licensing for dSPACE Products</b>	<b>33</b>
Main Benefits.....	34
Main Differences Between the Old and New Licensing Technologies.....	34
New Setup Program for Software Installation.....	36
License Activation.....	38
Decryption of Encrypted Archives of dSPACE Software.....	40
Compatibility Information.....	41
Requirements for Communication.....	42
<b>Aspects of Migrating from Previous Releases</b>	<b>45</b>
Migrating to dSPACE Release 2017-B.....	45
<b>AutomationDesk</b>	<b>47</b>
New Features of AutomationDesk 5.5.....	47
Migrating to AutomationDesk 5.5.....	51
<b>Automotive Simulation Models (ASM)</b>	<b>53</b>
All ASM Blocksets.....	54
New Features of All ASM Blocksets.....	54
Migrating All ASM Blocksets.....	54
ASM Brake Hydraulics Blockset.....	55
Migrating to ASM Brake Hydraulics Blockset 2.0.1.....	55

ASM Diesel Engine Blockset.....	56
New Features of ASM Diesel Engine Blockset 2.6.....	56
Changes in the ASM Diesel Engine Demo Model.....	59
Migrating to ASM Diesel Engine Blockset 2.6.....	60
ASM Diesel Exhaust Blockset.....	62
New Features of ASM Diesel Exhaust Blockset 2.1.5.....	62
Changes in the ASM Diesel Exhaust Demo Model.....	62
Migrating to ASM Diesel Exhaust Blockset 2.1.5.....	62
ASM Diesel InCylinder Blockset.....	63
Changes in the ASM Diesel InCylinder Demo Model.....	63
ASM Drivetrain Basic Blockset.....	64
New Features of ASM Drivetrain Basic Blockset 5.0.....	64
Migrating to ASM Drivetrain Basic Blockset 5.0.....	65
ASM Environment Blockset.....	66
New Features of ASM Environment Blockset 4.7.....	66
ASM Gasoline Engine Basic Blockset.....	67
Changes in the ASM Engine Gasoline Basic Demo Model.....	67
ASM Gasoline Engine Blockset.....	68
New Features of ASM Gasoline Engine Blockset 4.0.....	68
Changes in the ASM Engine Gasoline Demo Model.....	70
Migrating to ASM Gasoline Engine Blockset 4.0.....	70
ASM Gasoline InCylinder Blockset.....	72
Changes in the ASM Gasoline InCylinder Demo Model.....	72
ASM Traffic Blockset.....	73
New Features of ASM Traffic Blockset 3.7.....	73
Changes in the ASM Traffic Demo Model.....	73
Migrating to ASM Traffic Blockset 3.7.....	74
ASM Trailer Blockset.....	75
Changes in the ASM Trailer Demo Model.....	75
Migrating to ASM Trailer Blockset 2.6.2.....	75
ASM Truck Blockset.....	76
Changes in the ASM Truck Demo Model.....	76
Migrating to ASM Truck Blockset 3.0.2.....	76
ASM Vehicle Dynamics Blockset.....	77
New Features of ASM Vehicle Dynamics Blockset 4.0.....	77
Changes in the ASM Vehicle Dynamics Demo Model.....	77
Migrating to ASM Vehicle Dynamics Blockset 4.0.....	77

<b>Bus Manager (Stand-Alone)</b>	<b>79</b>
Features of the Bus Manager (Stand-Alone) 6.0.....	79
Migrating to Bus Manager (Stand-Alone) 6.0.....	81
<b>ConfigurationDesk</b>	<b>83</b>
ConfigurationDesk - Implementation Version.....	84
New Features of ConfigurationDesk 6.0 (Implementation Version).....	84
Migrating to ConfigurationDesk 6.0.....	92
ConfigurationDesk - Configuration Version.....	95
New Features of ConfigurationDesk 6.0 (Configuration Version).....	95
<b>ControlDesk</b>	<b>97</b>
New Features of ControlDesk 6.2.....	98
New Features of Platform Management and Platforms/Devices (ControlDesk 6.2).....	98
New Variable Management Features (ControlDesk 6.2).....	100
New Instrument Features (ControlDesk 6.2).....	102
New Measurement and Recording Features (ControlDesk 6.2).....	105
New Automation Features (ControlDesk 6.2).....	105
New Bus Navigator Features (ControlDesk 6.2).....	106
New Signal Editor Features (ControlDesk 6.2).....	109
Further Enhancements and Changes with ControlDesk (ControlDesk 6.2).....	110
Migrating to ControlDesk 6.2.....	113
Discontinuations in ControlDesk.....	113
Migrating to ControlDesk 6.2.....	114
<b>DCI Configuration Tool</b>	<b>121</b>
New Features of the DCI Configuration Tool 3.8.....	121
<b>dSPACE CAN API Package</b>	<b>123</b>
New Features of dSPACE CAN API Package 3.0.2.....	123
<b>dSPACE ECU Flash Programming Tool</b>	<b>125</b>
New Features of the dSPACE ECU Flash Programming Tool 2.4.....	125

<b>dSPACE FlexRay Configuration Package</b>	<b>127</b>
New Features of dSPACE FlexRay Configuration Package 4.0.....	127
<b>dSPACE XIL API .NET</b>	<b>129</b>
New Features of dSPACE XIL API .NET 2017-B.....	129
<b>ECU Interface Manager</b>	<b>131</b>
New Features of ECU Interface Manager 2.2.....	131
Migrating to ECU Interface Manager 2.2.....	132
<b>Model Compare</b>	<b>133</b>
New Features of Model Compare 2.8.....	133
Migration to Model Compare 2.8.....	135
<b>ModelDesk</b>	<b>137</b>
New Features of ModelDesk 4.6.....	137
Migration to ModelDesk 4.6.....	139
<b>Model Interface Package for Simulink</b>	<b>141</b>
New Features of the Model Interface Package for Simulink 3.5.....	141
Migration Aspects of the Model Interface Package for Simulink.....	144
<b>MotionDesk</b>	<b>147</b>
New Features of MotionDesk 4.1.....	147
Migrating to MotionDesk 4.1.....	149
<b>Real-Time Testing</b>	<b>151</b>
New Features of Real-Time Testing 3.3.....	151
Migrating to Real-Time Testing 3.3.....	152
<b>RTI/RTI-MP and RTLib</b>	<b>155</b>
New Features of RTI/RTI-MP and RTLib.....	155
Migration Aspects of RTI/RTI-MP and RTLib.....	156

<b>RTI Bypass Blockset</b>	<b>157</b>
New Features of the RTI Bypass Blockset 3.9.....	157
Migrating to RTI Bypass Blockset 3.9.....	157
<b>RTI CAN MultiMessage Blockset</b>	<b>159</b>
New Features of the RTI CAN MultiMessage Blockset 4.6.....	159
Migrating to RTI CAN MultiMessage Blockset 4.6.....	160
<b>RTI FPGA Programming Blockset</b>	<b>163</b>
New Features of the RTI FPGA Programming Blockset 3.4.....	163
Migrating to RTI FPGA Programming Blockset 3.4.....	165
<b>RTI LIN MultiMessage Blockset</b>	<b>167</b>
New Features of the RTI LIN MultiMessage Blockset 2.9.....	167
Migrating to RTI LIN MultiMessage Blockset 2.9.....	168
<b>SCALEXIO Firmware</b>	<b>169</b>
New Features of the SCALEXIO Firmware 4.1.....	169
<b>SYNECT</b>	<b>171</b>
New Features of SYNECT 2.4.....	172
New General Features of SYNECT.....	172
New Features of Test Management.....	176
New Features of Model and Signal & Parameter Management.....	178
New Features of Workflow Management.....	181
Migrating to SYNECT 2.4.....	183
Migrating Databases.....	183
<b>SystemDesk</b>	<b>185</b>
New Features of SystemDesk 5.0.....	186
New General Features.....	186
Configuring ECUs.....	188
Managing V-ECUs.....	189
Splittable AUTOSAR Elements.....	190
Process Support With AUTOSAR Master Files.....	192

Migrating to SystemDesk 5.0.....	193
Migrating to SystemDesk 5.0.....	193
<b>TargetLink</b>	<b>195</b>
New Features of TargetLink 4.3 and TargetLink Data Dictionary 4.3.....	196
Modeling in Simulink or Stateflow.....	196
Newly Supported Simulink Block.....	197
Bus-Capable Custom Code Block (Type II).....	197
Other Simulink/Stateflow Features.....	197
Code Generation Core Functionality.....	198
Code Decorations (Declaration Statements and Section Names).....	199
MISRA C Compliance.....	199
Improved Support of Variable Vector Widths.....	200
Modular Development.....	201
Improved A2L File Generation.....	201
Improved Workflow for Distributed Development.....	202
AUTOSAR.....	202
Supported AUTOSAR Releases.....	202
Memory Mapping.....	203
Static Memories and Constant Memories for Measurement and Calibration.....	203
Support for Rte_IsUpdated.....	204
AUTOSAR Import and Export Improvements.....	204
Target Simulation (PIL).....	205
Changes in the Target Simulation Modules.....	205
Data Dictionary and Data Management.....	206
Further Improvements to the Data Dictionary.....	206
Code Generator Options.....	206
New Code Generator Options.....	206
API Functions and Hook Scripts.....	207
New API Functions.....	207
Other.....	207
New Property Manager.....	208
General Enhancements and Changes.....	209
TargetLink Demos.....	210
Migrating to TargetLink 4.3 and TargetLink Data Dictionary 4.3.....	212
MATLAB-Related Changes.....	212



General Migration Information.....	213
Upgrade of Models, Libraries, and Data Dictionaries.....	213
Basics on Migrating between TargetLink Versions.....	213
How to Upgrade a Data Dictionary with Included DD Files.....	216
How to Manually Upgrade Libraries and Models via the API.....	218
Migrating Data Dictionaries to CodeDecorationSets.....	219
Migrating from TargetLink 4.2 to 4.3.....	222
Code Generator Options.....	222
Migration Aspects Regarding Code Generator Options.....	222
API Functions and Hook Scripts.....	225
Changes in TargetLink and TargetLink Data Dictionary API Functions.....	225
AUTOSAR-Related Migration Aspects.....	226
Other.....	226
Property Manager.....	226
Various Migration Aspects.....	228
Code Changes.....	232
64-Bit Multiplication.....	232
AUTOSAR.....	234
Efficiency.....	236
Function Reuse.....	240
MISRA Compliance.....	241
Mixed Operations (Floating-Point and Fixed-Point Types).....	242
State Reset.....	244
Other.....	250
Discontinuations.....	255
Discontinued TargetLink Features .....	255
Obsolete API Functions.....	256
Obsolete Limitations.....	256
Changes in Future TargetLink Versions.....	258
Features to Be Discontinued.....	258
API Functions to Be Discontinued.....	259
Deprecated Code Generator Options.....	260
Other.....	260
<b>VEOS</b> .....	<b>261</b>
New Features of VEOS 4.1.....	261
Compatibility of VEOS 4.1.....	262

Migrating to VEOS 4.1.....	264
Discontinuations in VEOS.....	265
<b>Compatibility Information</b>	<b>267</b>
Supported MATLAB Releases.....	267
Operating System.....	268
Notes on 64-Bit Compatibility of dSPACE Products .....	270
Run-Time Compatibility of dSPACE Software.....	270
Limitations for Using Windows Features.....	271
<b>Index</b>	<b>275</b>

# About This Document

## Contents

This document informs you about the new features of all the dSPACE software products in Release 2017-B. It also gives you an overview of software products with no or minor changes. There are instructions on migrating from earlier dSPACE releases, especially from earlier product versions, if required.

## Where to go from here





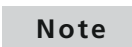

### Information in this section



Conventions Used in dSPACE User Documentation .....	11
Accessing dSPACE Help and PDF Files .....	12

## Conventions Used in dSPACE User Documentation

### Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
 <b>DANGER</b>	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
 <b>WARNING</b>	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
 <b>CAUTION</b>	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
 <b>NOTICE</b>	Indicates a hazard that, if not avoided, could result in property damage.
 <b>Note</b>	Indicates important information that you should take into account to avoid malfunctions.
 <b>Tip</b>	Indicates tips that can make your work easier.

Symbol	Description
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

### Naming conventions

dSPACE user documentation uses the following naming conventions:

**%name%** Names enclosed in percent signs refer to environment variables for file and path names.

**< >** Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

### Special folders

Some software products use the following special folders:

**Common Program Data folder** A standard folder for application-specific configuration data that is used by all users.

`%PROGRAMDATA%\dSPACE\\`

or

`%PROGRAMDATA%\dSPACE\\`

**Documents folder** A standard folder for user-specific documents.

`%USERPROFILE%\My Documents\dSPACE\\`

**Local Program Data folder** A standard folder for application-specific configuration data that is used by the current, non-roaming user.

`%USERPROFILE%\AppData\Local\dSPACE\\`

## Accessing dSPACE Help and PDF Files

### Introduction

After you install and decrypt your dSPACE software, the documentation for the installed products is available as online help in dSPACE Help and as Adobe® PDF files.

## Online help

There are various ways to open dSPACE Help.

### Note

Not all the ways to open dSPACE Help are available for all dSPACE software products.


**Opening from Windows** You can open dSPACE Help on its home page:

- Via Windows Start Menu

**Opening from dSPACE software with menu bar** You can open dSPACE Help on a product's start page:

- Via the menu bar in a dSPACE product

**Opening from dSPACE software with ribbons** If you use dSPACE software with ribbons, you can open dSPACE Help:


- Via the **Start** page in dSPACE software
- Via the Backstage view in dSPACE software (leftmost ribbon tab)
- Via the  button

**Opening context-sensitive help** dSPACE Help provides context-sensitive help. You can open help on the currently active context in dSPACE software:

- Via **F1**
- Via the **Help** button


## PDF files

You can open the PDF files as follows:

**Opening from a topic in dSPACE Help** You can access the PDF file with the current topic via the  button at the topic's top right. The following illustration shows an example:



The PDF document opens on its first page.

**Opening from dSPACE software with ribbons** If your dSPACE software has a user interface with ribbons, you can open a folder containing the user documentation in PDF format via the  button in the Backstage view (leftmost ribbon tab).



# Overview of dSPACE Release 2017-B

**Introduction** Gives you an overview of the new key features in Release 2017-B and information about unchanged products.

**Where to go from here**

**Information in this section**

General Enhancements and Changes .....	15
New dSPACE Help Available .....	17
Product Version Overview .....	21
New Product Key Features .....	23

## General Enhancements and Changes

**Introduction** The following new features and changes concern several dSPACE products.

**64-bit Python distribution** dSPACE Release 2017-B contains a 64-bit Python distribution with the following components.

Python Component	64-Bit Version
Python core	2.7.13
PyWin32	221.10
Numpy	1.12.1
Matplotlib	1.5.3
WxPython	3.0.2.0
Py2exe	0.6.9
Comtypes	1.1.3

Python Component	64-Bit Version
Python for .NET	2.3.0
Cycler	0.10.0
Pillow	4.1.1
Pip	9.0.1
Pyparsing	2.2.0
Python_dateutil	2.6.0
Pytz	2017.2
Six	1.10.0

**RCP and HIL software: C/C++ compiler for building MATLAB MEX files change with dSPACE Release 2017-B**

RCP and HIL software (such as RTI CAN MultiMessage Blockset, RTI LIN MultiMessage Blockset, or Automotive Simulation Models) now supports the following C/C++ compiler to build MATLAB MEX files:

- MinGW (GNU Compiler Collection (GCC 4.9.2)): In combination with MATLAB Releases 2016a, 2016b and 2017a.
- MinGW (GNU Compiler Collection (GCC 5.3.0)): In combination with MATLAB Release 2017b.
- Microsoft Visual Studio 2015 Pro: In combination with MATLAB Release 2016a, 2016b, 2017a and 2017b.

**Note**

The Microsoft Windows SDK 7.1 C/C++ compiler is no longer supported.

**New licensing technology and setup program**

dSPACE Release 2017-B introduces a new licensing technology. An improved setup program for installing dSPACE software is now available. For more information, refer to [New Licensing for dSPACE Products](#) on page 33.

**New help**

As of dSPACE Release 2017-B, dSPACE HelpDesk is discontinued and replaced by dSPACE Help. For more information, refer to [New dSPACE Help Available](#) on page 17.

**New PDF file layout**

The PDF files of dSPACE Release 2017-B introduce a new layout. The layout is designed to provide information more clearly and has been harmonized with the layout of the new help. The paper size of PDF files has been changed to better support printing.



---

**Printed user documentation**

The printed user documentation is not delivered automatically. You can decide which of the available printed documents you would like to have. To order printed documentation, refer to <http://www.dspace.com/go/requestreleasematerial>.

**Note**

If you do not order printed documentation, use dSPACE Help or PDF files to learn about new features, enhancements, and the safety precautions regarding your products.

---

**Discontinuation of dSPACE hardware**

**DCI-CAN1** This product will be discontinued as of December 2017. New Releases of dSPACE software will continue to support the DCI-CAN1 at least until the end of 2019.

However, for new projects we recommend that you use the successors DCI-CAN2 or DCI-CAN/LIN1.

**DCI-GSI1** This product will be discontinued as of December 2017. New Releases of dSPACE software will continue to support the DCI-GSI1 at least until the end of 2019.

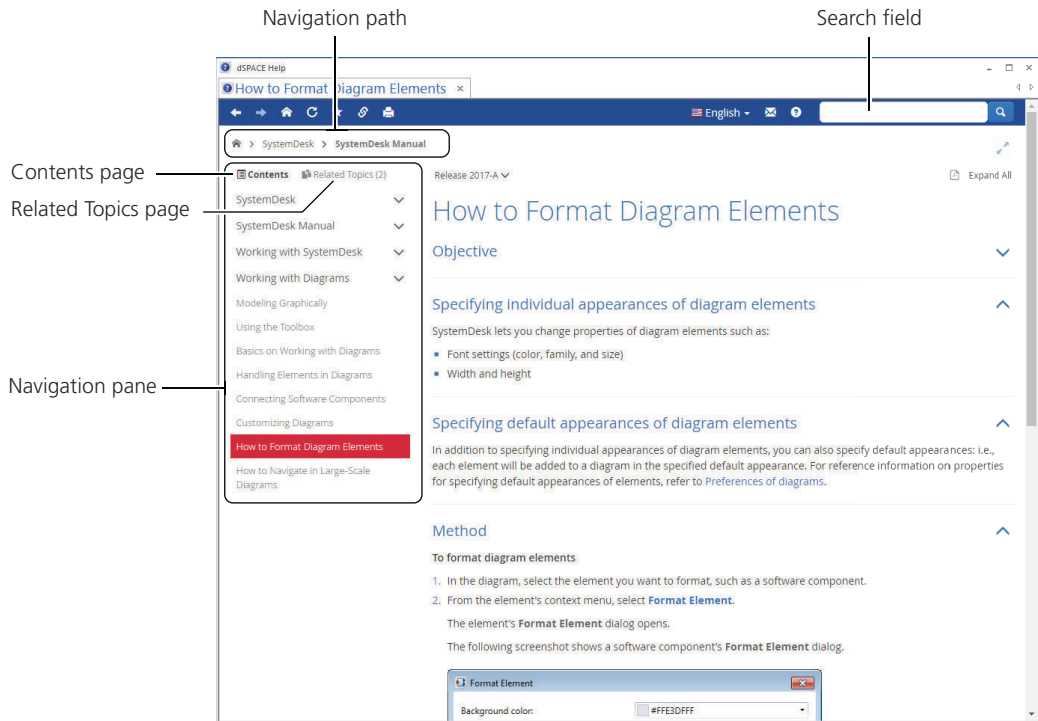
However, for new projects we recommend that you use the successor DCI-GSI2.

## New dSPACE Help Available

---

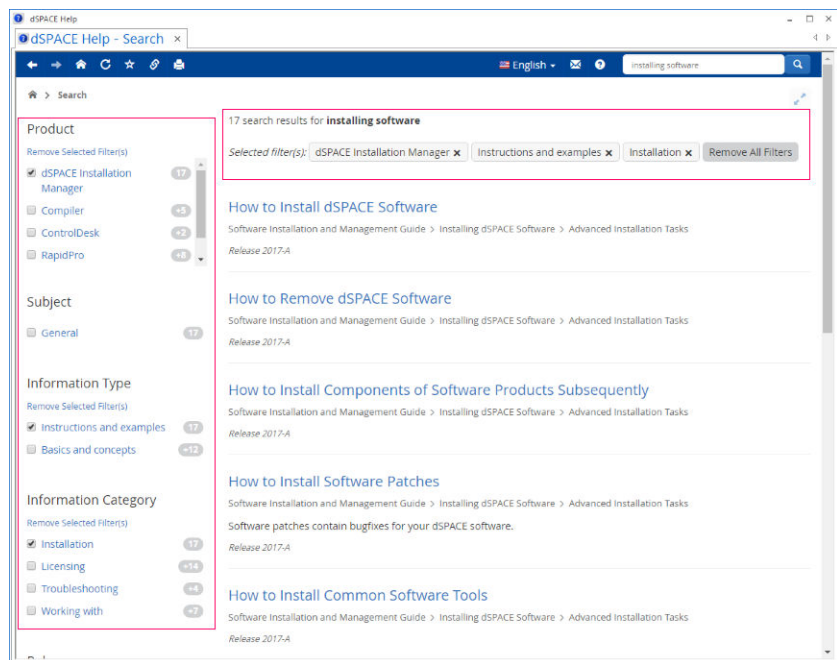
**dSPACE help**

With Release 2017-B dSPACE provides the new dSPACE Help. It offers you some helpful new features. The key benefits are listed in this topic. dSPACE Help is based on the software Acolada Cobrili.

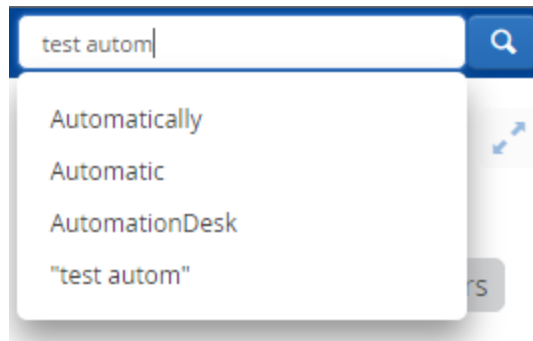


## Search

**Faceted filtering** Faceted filtering enables you to narrow down the results of a full-text-search according to predefined filters such as *Product*, *subject*, and *information category*. The facets show the number of the remaining results. The following illustration shows the search results page with applied filters.



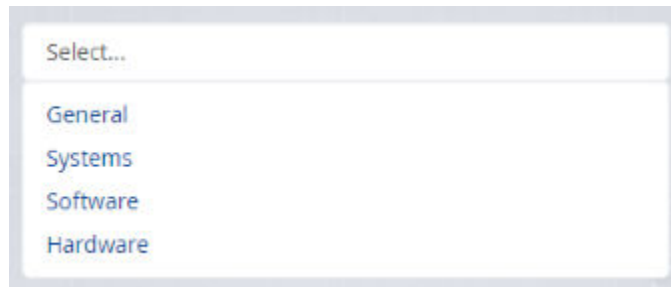
**Search suggestions** Entering search queries is now supported by suggestions. This helps you to search for exact words and to get better results.



## Navigation


dSPACE Help provides several navigation options. The documentation selector helps you navigate to a specific product documentation. From there, a combination of navigation path and main navigation guides you through the documentation structures.

**Documentation selector** The *documentation selector* is a new feature of dSPACE Help. It helps you navigate through the documentation structure if several products, product versions or dSPACE Releases are installed on your PC.



**Main navigation** The main navigation is located on the left side in the navigation pane. It is hierarchically structured and shows the selected topic, the sibling topics, and all sections located on higher levels up to the home page.

**Navigation path** The navigation path, also known as bread crumb trail, helps you keep overview while navigating. It is located in the header area and consists of links to the higher hierarchy levels.

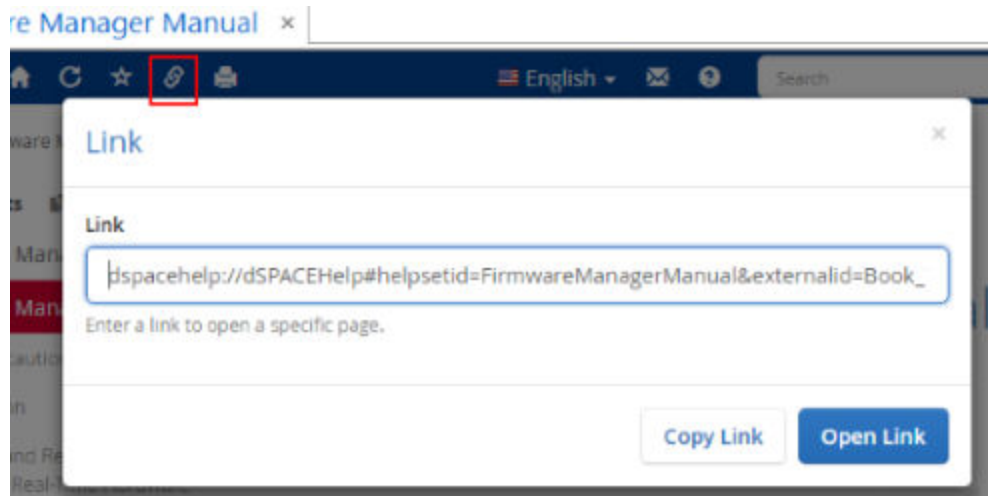
**Further navigation options** You have still the opportunity to navigate to the last viewed topics by using the  buttons in the top menu bar. The functionality is similar to standard internet browsers.

You can also navigate hierarchically through a documentation structure by using the previous and next links at the end of every page.

## Link sharing

dSPACE Help lets you copy and paste links to a specific page. This feature makes it easy for you to communicate with colleagues or dSPACE Support. Sending and

receiving links via emails, internet forms or other communication platforms enables you to share information in dSPACE Help quickly and exactly.



### User interface

**Browser handling** dSPACE Help is based on standard web browser technology. In addition to familiar controls in the menu bar that are also featured in other browsers, the help provides the following features:

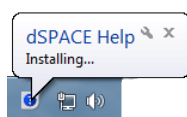
- Tabbed browsing
- Frequently used shortcuts
- Expanding and collapsing of paragraphs
- Switching between languages
- Switching between installed Releases

**Responsive design** The user interface of dSPACE Help is responsive, this means, if you change the size of the window, the content and the control elements will be adjusted automatically for a correct screen presentation.

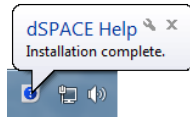
**Optimized clarity and legibility** *Related Topics* are now placed in the navigation pane, so they are more visible and accessible. In addition to the optimized on-screen-presentation of the content, clarity and legibility were also improved.

### Visualization of the installation process

If you use dSPACE Help directly after installation, the content might be incomplete, because the installation of the help application and the installation of the documentation data are separate steps. The installation time of the content depends on the volume and can take a few minutes. A tray icon and notifications in the Windows task bar will inform you about the installation progress so you can see when new content is completely available.




When the installation has finished the following notification is displayed.



#### Note

As of Release 2017-B, some dSPACE products and their documentation are encrypted on delivery. Before the documentation can be installed and displayed in dSPACE Help you have to decrypt it using dSPACE Installation Manager. For more information of decrypting, refer to [Decryption of Encrypted Archives of dSPACE Software](#) on page 40.

#### Using dSPACE Help

For a more detailed description of the features and of how to use the help, click  in dSPACE Help.

## Product Version Overview

#### Product versions

The following table is an extract from product version histories showing the product versions of the current Release and of the three preceding Releases. If a product has new features, there is a link to the brief description in this document.

Product	dSPACE Release			
	2016-A	2016-B	2017-A	2017-B
AutomationDesk	5.2	5.3	5.4	5.5 Refer to <a href="#">AutomationDesk</a> on page 47.
Automotive Simulation Models	8.2	8.3	8.4	9.0 Refer to <a href="#">Automotive Simulation Models (ASM)</a> on page 53.
Bus Manager (stand-alone)	5.5	5.6	5.7	6.0 Refer to <a href="#">Bus Manager (Stand-Alone)</a> on page 79.
ConfigurationDesk	5.5	5.6	5.7	6.0 Refer to <a href="#">ConfigurationDesk</a> on page 83.
Container Manager	4.4	4.5	4.5	5.0
ControlDesk	5.6	6.0	6.1	6.2 Refer to <a href="#">ControlDesk</a> on page 97.
DCI Configuration Tool	3.6	3.7	3.7.1	3.8 Refer to <a href="#">DCI Configuration Tool</a> on page 121.

Product	dSPACE Release			
	2016-A	2016-B	2017-A	2017-B
dSPACE CAN API Package	2.7.5	3.0	3.0.1	3.0.2 Refer to <a href="#">dSPACE CAN API Package</a> on page 123.
dSPACE ECU Flash Programming Tool	2.3	2.3.1	2.3.2	2.4 Refer to <a href="#">dSPACE ECU Flash Programming Tool</a> on page 125.
dSPACE FlexRay Configuration Package	3.7	3.8	3.9	4.0 Refer to <a href="#">dSPACE FlexRay Configuration Package</a> on page 127.
dSPACE HIL API .NET	2.1	-	-	-
dSPACE Python Extensions	2.1	2.2	2.3	2.4
dSPACE XIL API .NET	2016-A	2016-B	2017-A	2017-B Refer to <a href="#">dSPACE XIL API .NET</a> on page 129.
ECU Interface Manager	1.8	2.0	2.1	2.2 Refer to <a href="#">ECU Interface Manager</a> on page 131.
Firmware Manager	2.1	2.2	2.3	2.4
Model Compare	2.6	2.7	2.7	2.8 Refer to <a href="#">Model Compare</a> on page 133.
ModelDesk	4.3	4.4	4.5	4.6 Refer to <a href="#">ModelDesk</a> on page 137.
Model Interface Package for Simulink	3.2	3.3	3.4	3.5 Refer to <a href="#">Model Interface Package for Simulink</a> on page 141.
MotionDesk	3.8	3.9	4.0	4.1 Refer to <a href="#">MotionDesk</a> on page 147.
MotionDesk Blockset	2.4.1	2.5	2.5.1	2.5.2 Refer to <a href="#">MotionDesk</a> on page 147.
Real-Time Testing	3.0	3.1	3.2	3.3 Refer to <a href="#">Real-Time Testing</a> on page 151.
RTI <sup>1)</sup>	7.6	7.7	7.8	7.9 Refer to <a href="#">RTI/RTI-MP and RTLib</a> on page 155.
RTI-MP <sup>2)</sup>	7.6	7.7	7.8	7.9 Refer to <a href="#">RTI/RTI-MP and RTLib</a> on page 155.
RTI Bypass Blockset	3.6	3.7	3.8	3.9 Refer to <a href="#">RTI Bypass Blockset</a> on page 157.
RTI CAN Blockset	3.4.2	3.4.3	3.4.4	3.4.5
RTI CAN MultiMessage Blockset	4.3	4.4	4.5	4.6 Refer to <a href="#">RTI CAN MultiMessage Blockset</a> on page 159.
RTI Electric Motor Control Blockset	1.3	1.3.1	1.4	1.4.1
RTI Ethernet Blockset	1.2	1.2.1	1.2.2	1.2.3
RTI Ethernet (UDP) Blockset	1.4	1.4.1	1.4.2	1.4.3

Product	dSPACE Release			
	2016-A	2016-B	2017-A	2017-B
RTI FPGA Programming Blockset	3.1	3.2	3.3	3.4 Refer to <a href="#">RTI FPGA Programming Blockset</a> on page 163.
RTI LIN MultiMessage Blockset	2.6	2.7	2.8	2.9 Refer to <a href="#">RTI LIN MultiMessage Blockset</a> on page 167.
RTI RapidPro Control Unit Blockset	2.2.1	2.2.2	2.2.2	2.2.3
RTI USB Flight Recorder Blockset	1.2	1.2.1	1.2.1	1.2.2
RTI Watchdog Blockset	1.0	2.0	2.1	2.1.1
SCALEXIO firmware	3.4	3.5	4.0	4.1 Refer to <a href="#">SCALEXIO Firmware</a> on page 169.
SYNECT	2.1	2.2	2.3	2.4 Refer to <a href="#">SYNECT</a> on page 171.
SystemDesk	4.6	4.7	4.8	5.0 Refer to <a href="#">SystemDesk</a> on page 185.
TargetLink/ <b>TargetLink Data Dictionary</b>	4.1	4.2	4.2	4.3 Refer to <a href="#">TargetLink</a> on page 195.
Variable Editor	2.3	2.3 <sup>3)</sup>	2.3 <sup>3)</sup>	2.4 <sup>3)</sup>
VEOS	3.6	3.7	4.0	4.1 Refer to <a href="#">VEOS</a> on page 261.

<sup>1)</sup> Including the standard I/O blocksets.

<sup>2)</sup> Including the RTI Gigalink Blockset.

<sup>3)</sup> The Variable Editor is no longer part of the dSPACE Release DVD. It is available at <https://www.dspace.com/go/requestreleasedownload>.

If you have not performed regular updates, refer to the *New Features and Migration* documents for the dSPACE Releases listed above for information about the new features and necessary migration steps.

## New Product Key Features

### Introduction

This is an overview of each product's new key features. For more information, refer to the product-specific sections.

### Where to go from here

#### Information in this topic

<a href="#">AutomationDesk</a> .....	24
<a href="#">Bus Manager (stand-alone)</a> .....	24
<a href="#">ConfigurationDesk (Implementation Version)</a> .....	25
<a href="#">ConfigurationDesk (Configuration -Version)</a> .....	25

ControlDesk .....	25
DCI Configuration Tool .....	26
dSPACE CAN API Package .....	27
dSPACE ECU Flash Programming Tool .....	27
dSPACE FlexRay Configuration Package .....	27
dSPACE XIL API .....	27
ECU Interface Manager .....	27
Firmware Manager .....	27
ModelDesk .....	28
MotionDesk .....	28
Python Extensions .....	28
Real-Time Testing .....	28
RTI, RTI-MP, and RTLib .....	28
RTI CAN MultiMessage Blockset .....	29
RTI FPGA Programming Blockset .....	29
RTI LIN MultiMessage Blockset .....	29
SCALEXIO firmware .....	29
SYNECT .....	29
SystemDesk .....	30
TargetLink .....	30
VEOS .....	30

**AutomationDesk**

The new key features of AutomationDesk are:

- Support of the XIL API framework concept for centrally managing ports and variables, featuring new configuration dialogs, new automation blocks, and new data objects.
- Updated Mapping Editor to support the new requirements of the XIL API Framework.
- Enhancements to the Signal Editor, such as the handling of STZ files.
- Enhanced error handling for fixed parameters when using them in the context of the Signal Editor, the XIL API Framework mapping, and other XIL API use scenarios.
- Enhancements to the electrical error simulation, such as the support of the DS5390 High Current FIU.
- Enhancements to the Main Library, such as the new Bool data object and the attachment feature of the File data object.
- Enhancements to the Report library, such as the new AddHTML block.
- Enhanced handling of data objects within the Sequence Builder.
- Enhancements to the COM API, such as accessing the library path.

For more information on the new features, refer to [New Features of AutomationDesk 5.5](#) on page 47.

**Bus Manager (stand-alone)**

The new key features of the Bus Manager (stand-alone) are:

- Enhanced user interface
- New bus configuration features
- Additional configurable communication matrix elements



- New PDU elements to access supported PDU types
- Enhanced bus configuration tables

For more information, refer to [Features of the Bus Manager \(Stand-Alone\) 6.0](#) on page 79.

---

### ConfigurationDesk (Implementation Version)

The new key features of ConfigurationDesk are:

- The user interface of ConfigurationDesk has been improved to support your tasks and use scenarios more efficiently. ConfigurationDesk now offers different view sets that provide panes and commands suited for specific use scenarios and purposes. You can switch between view sets by using the navigation bar. For each view set, the **Home** ribbon contains specific commands suitable for the purpose of the view set.
- ConfigurationDesk now provides the **Model-Function Mapping Browser**. The **Model-Function Mapping Browser** displays all the model implementations including their subsystems of the active ConfigurationDesk application. You can easily create signal chains by dragging and dropping hardware channels or function blocks to a Simulink behavior model or its subsystem.
- Support of new SCALEXIO hardware: DS6202 Digital I/O Board, DS6311 FlexRay Board, DS6341 CAN Board, DS6351 LIN Board, and DS6551 IOCNET Link Board.
- Support of new function block types: Digital Incremental Encoder In, Multi-Channel PWM Out, and Digital Pulse Out.
- Various enhancements of the Bus Manager for configuring bus communication for simulation and inspection purposes.
- Support of ECU calibration page handling and data access to individual ECU variables for ECU interfacing with SCALEXIO systems.

For more information, refer to [ConfigurationDesk - Implementation Version](#) on page 84.

---

### ConfigurationDesk (Configuration -Version)

The new key feature of ConfigurationDesk is:

- The user interface is more intuitive: Its menu bar and toolbars have been replaced by ribbons.

---

### ControlDesk

The new key features of ControlDesk 6.2 are:

#### Platform/device enhancements

- New Ethernet Bus Monitoring device
- XIL API MAPort platform: Example showing the access to third-party hardware
- ECU Diagnostics device: Support of CAN channels of dSPACE SCALEXIO and VEOS

For more information on the new features, refer to [New Features of Platform Management and Platforms/Devices \(ControlDesk 6.2\)](#) on page 98.

#### Variable management enhancements

- A2L file import: Ignoring variables at the address 0x0
- AUTOSAR/FIBEX file import: Support of multi-cluster files
- Display of variable properties in the **Properties** controlbar
- Support of Int64 and UInt64 variables in TRC files

For more information on the new features, refer to [New Variable Management Features \(ControlDesk 6.2\)](#) on page 100.

#### Instrument enhancements

- New Map instrument
- Table Editor: Connecting multidimensional table data (n-D tables)
- Time Plotter/Index Plotter: Support of variables using conversion tables

For more information on the new features, refer to [New Instrument Features \(ControlDesk 6.2\)](#) on page 102.

#### Measurement and recording enhancements

- Support of variable-specific default rasters

For more information on the new features, refer to [New Measurement and Recording Features \(ControlDesk 6.2\)](#) on page 105.

#### Bus Navigator enhancements

- Ethernet monitoring
- Ethernet capture filter
- Logging bus statistics
- AUTOSAR/FIBEX file import: Support of multicluster files
- AUTOSAR file import: Support of AUTOSAR 4.3.0
- Monitoring list: Column configuration via column sets

For more information on the new features, refer to [New Bus Navigator Features \(ControlDesk 6.2\)](#) on page 106.

#### Signal Editor enhancements

- Highlighting mapped signals

For more information on the new features, refer to [New Signal Editor Features \(ControlDesk 6.2\)](#) on page 109.

#### Automation enhancements

- Adding/removing parameters to/from a sub data set

For more information on the new features, refer to [New Automation Features \(ControlDesk 6.2\)](#) on page 105.

#### Further enhancements

- Filtering items in the **Properties** controlbar

For more information on the new features, refer to [Further Enhancements and Changes with ControlDesk \(ControlDesk 6.2\)](#) on page 110.

---

#### DCI Configuration Tool

The new key feature of the DCI Configuration Tool is:

- Improved A2L file adaptation

For more information on the new feature, refer to [New Features of the DCI Configuration Tool 3.8](#) on page 121.

---

**dSPACE CAN API Package**

The new key feature of the dSPACE CAN API Package is:

- Checking the availability of a specific CAN channel (dSPACE CAN API 2.0 only)

For more information on the new features, refer to [New Features of dSPACE CAN API Package 3.0.2](#) on page 123.

---

**dSPACE ECU Flash Programming Tool**

The new key feature of the dSPACE ECU Flash Programming Tool is:

- Associations of project files and ECU data files with the dSPACE ECU Flash Programming Tool

For more information on the new feature, refer to [New Features of the dSPACE ECU Flash Programming Tool 2.4](#) on page 125.

---

**dSPACE FlexRay Configuration Package**

The new key feature of the dSPACE FlexRay Configuration Package is:

- Support of SCALEXIO systems with a DS6311 FlexRay Board

For more information on the new features, refer to [New Features of dSPACE FlexRay Configuration Package 4.0](#) on page 127.

---

**dSPACE XIL API**

The new key features of dSPACE XIL API are:

- XIL API MAPort now supports scaling of stimulus signals.
- XIL API EESPort now supports the dSPACE CAN API version 2.0, and custom properties.

For more information on the new features, refer to [New Features of dSPACE XIL API .NET 2017-B](#) on page 129.

---

**ECU Interface Manager**

The new key features of the ECU Interface Manager are:

- ECU calibration page handling via XCP
- Data access configuration for ECU variables

For more information on the new features, refer to [New Features of ECU Interface Manager 2.2](#) on page 131.

---

**Firmware Manager**

The new key feature of the Firmware Manager is:

- Support of new SCALEXIO boards:
  - DS6202 Digital I/O Board
  - DS6311 FlexRay Board
  - DS6341 CAN Board
  - DS6351 LIN Board

For more information on the new hardware, refer to [New Features of the SCALEXIO Firmware 4.1](#) on page 169.

---

### ModelDesk

The new key features of ModelDesk are:

- Creating a new ModelDesk project by selecting an ASM demo model.
- Adding subnodes to the Parameter Sets node of the project tree.
- Specifying traffic lights and sign gantries that can be controlled by the simulation.
- New sensor for traffic signs.
- Specifying trajectories
- Guard rails that can be evaluated in the simulation
- New steering mode for maneuver.
- Usability improvements of the Traffic Editor.
- Specifying downsampling factor for plotting.

For more information on the new features, refer to [New Features of ModelDesk 4.6](#) on page 137.

---

### MotionDesk

The new key features of MotionDesk are:

- Endless ground and sky as environment for the scene.
- Supporting very large networks.
- Advanced lighting mode that improves the lighting effects in the scene, so it looks more realistic.

For more information on the new features, refer to [New Features of MotionDesk 4.1](#) on page 147.

---

### Python Extensions

Python Extensions 2.4 has no new features.

---

### Real-Time Testing

The new key features of Real-Time Testing are:

- Supporting the scaling of the variables in the variable description file.
- The `rtlib.watcherlib` module supports the implementation of a watcher function in the RTT sequence.
- The `rtlib.canapilib` module supports bus statistics.

For more information on the new features, refer to [New Features of Real-Time Testing 3.3](#) on page 151.

---

### RTI, RTI-MP, and RTLib

The new key feature of RTI, RTI-MP, and RTLib is:

- Support of MATLAB R2017b

For more information on the new feature, refer to [New Features of RTI/RTI-MP and RTLib](#) on page 155.

---

**RTI CAN MultiMessage Blockset**

The new key features of the RTI CAN MultiMessage Blockset are:

- Support of SCALEXIO systems with a DS6341 CAN Board
- Support of AUTOSAR System Template 4.3.0
- Support of AUTOSAR E2E protection profiles 05 and 06

For more information on the new features, refer to [New Features of the RTI CAN MultiMessage Blockset 4.6](#) on page 159.

---

**RTI FPGA Programming Blockset**

The new key features of the RTI FPGA Programming Blockset are:

- Extended Xilinx® software support.
- Enhancements to the FPGA framework for MicroLabBox.
- New FPGA framework for the DS2655 (7K410) FPGA Base Board.
- Extended FPGA access with your experiment software for SCALEXIO systems.
- New UART demo model for SCALEXIO systems.

For more information on the new features, refer to [New Features of the RTI FPGA Programming Blockset 3.4](#) on page 163.

---

**RTI LIN MultiMessage Blockset**

The new key features of the RTI LIN MultiMessage Blockset are:

- Support of SCALEXIO systems with a DS6351 LIN Board
- Support of AUTOSAR System Template 4.3.0

For more information on the new features, refer to [New Features of the RTI LIN MultiMessage Blockset 2.9](#) on page 167.

---

**SCALEXIO firmware**

The new key features of the SCALEXIO firmware are:

- Support of the DS6202 Digital I/O Board
- Support of the DS6311 FlexRay Board
- Support of the DS6341 CAN Board
- Support of the DS6351 LIN Board
- Support of the DS6551 IOCNET Link Board

For more information on the new features, refer to [New Features of the SCALEXIO Firmware 4.1](#) on page 169.

---

**SYNECT**

The new key feature of SYNECT 2.4 is:

- Support for integrating system models and building them for virtual validation with VEOS.

For more information on the new features, refer to [New Features of SYNECT 2.4](#) on page 172.

## SystemDesk

The new key features of SystemDesk 5.0 are:

- The user interface of SystemDesk is now more intuitive: Its menu bar and toolbar have been replaced by *ribbons*.
- Improved support for basic software modules.
- Support for import scenarios that involve AUTOSAR splittable elements.

For more information on the new features, refer to [New Features of SystemDesk 5.0](#) on page 186.

---

## TargetLink

The new key features of TargetLink are:

- AUTOSAR
  - Revision 4.3.0 support
  - Memory mapping
  - Static memory and constant memory support
  - `Rte_IsUpdated` support
  - Import and export of array of struct data types
- Block-specific improvements
  - Support for Simulink's Delay block
  - Support for buses and structs for **Custom Code** (Type II) block
- Data Dictionary
  - Exporting incremental A2L files
  - Support of Version 1.6.1 of ASAM MCD-2 MC
- Completely revised TargetLink Property Manager
- Workflow improvements for modular development
- Improved MISRA C compliance for the generated production code and the Fixed-Point Library
- Miscellaneous
  - Improvements for variable vector widths
  - Better support of resettable subsystems and state reset
  - New progress bar for code generation

For more information on all the new features, refer to [New Features of TargetLink 4.3](#) and [TargetLink Data Dictionary 4.3](#) on page 196.

For more information on the TargetLink migration aspects (TargetLink, TargetLink AUTOSAR module, TargetLink Data Dictionary), refer to [Migrating to TargetLink 4.3](#) and [TargetLink Data Dictionary 4.3](#) on page 212.

---

## VEOS

The new key features of VEOS are:

- Import of SMC files created with SYNECT
- Support for further compiler versions
- Undoing/Redoing the most recent actions
- Customizing the screen arrangement

For more information on the new features, refer to [New Features of VEOS 4.1](#) on page 261.





# New Licensing for dSPACE Products

## Motivation

As of dSPACE Release 2017-B, the licensing technology for protecting dSPACE software changes. From this Release forward, dSPACE licensing is based on the CodeMeter licensing technology from Wibu-Systems.

The former licensing technologies, which are based on WibuKey (for dongle licenses) and FlexNet from Flexera (for floating network licenses), were used for 20 years without any modifications.

Switching to the CodeMeter licensing technology provides state-of-the-art license management and supports new business models and licensing features.

CodeMeter is a future-proof licensing technology from Wibu-Systems. It provides easy and more flexible license handling, and reduces the manual effort for customers as well as for dSPACE. To help customers fully benefit from these advantages, dSPACE also improves its license policy and dSPACE software installation.

## Where to go from here

### Information in this section

Main Benefits .....	34
Main Differences Between the Old and New Licensing Technologies .....	34
New Setup Program for Software Installation .....	36
License Activation .....	38
Decryption of Encrypted Archives of dSPACE Software .....	40
Compatibility Information .....	41
Requirements for Communication .....	42

## Main Benefits

---

### Benefits and features

The main benefits of using the CodeMeter licensing technology are:

- Easier and more flexible license management.
  - Central license management via dSPACE Installation Manager, for example, activating, updating and deactivating licenses.
  - Dongle licenses and floating network licenses are based on the same licensing technology (one supplier instead of two).
  - CodeMeter licenses are Release-independent and can be valid for various dSPACE Releases. As a result, the time-consuming license update process for customers with a Software Maintenance Service (SMS) contract becomes much easier. Furthermore, for SMS customers the frequency of license updates is reduced significantly.
  - License updates are performed almost completely automatic.
  - License information for all dSPACE products and versions is no longer distributed over many files on your host PC, but stored in one license container (for example, in a CmDongle) and thus easily portable.
  - Licensing becomes more transparent because the symmetry between product information and licenses increases: One product in the dSPACE Catalog has only one license code. In addition, licenses are given traceable names that are related to the product.
  - Planned for future Releases: Managing licenses via a Web portal, for example, to obtain a clear overview of the licenses available in your department.
- License-independent software installation.

The software installation process is now license-independent and therefore becomes simpler. No license files are required to install dSPACE software products.
- Support of new license features and business models.

Other features and models, such as license borrowing and pay-per-use, can be implemented in the future.

## Main Differences Between the Old and New Licensing Technologies

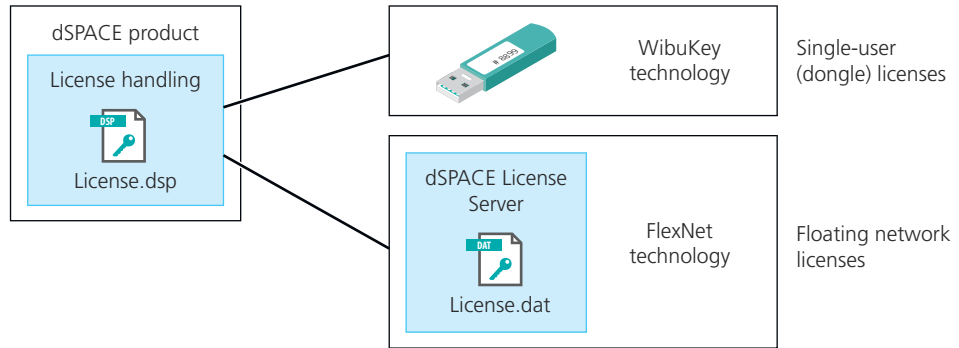
---

### Supported license types

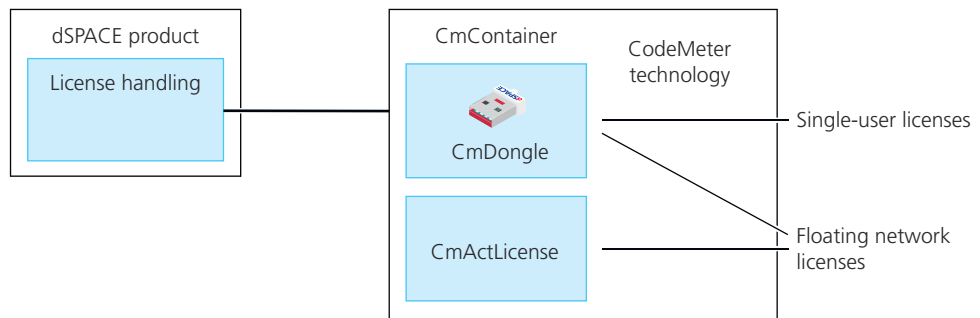
dSPACE software supports single-user licenses and floating network licenses.

**Main changes in technology**

Up to dSPACE Release 2017-A

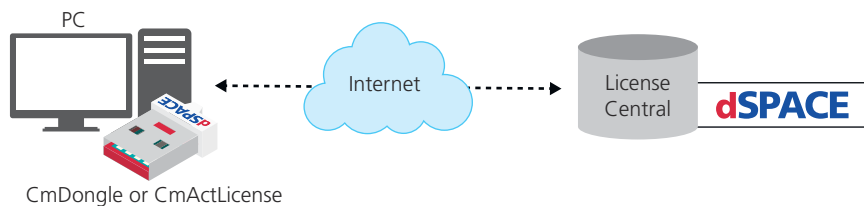


As of dSPACE Release 2017-B



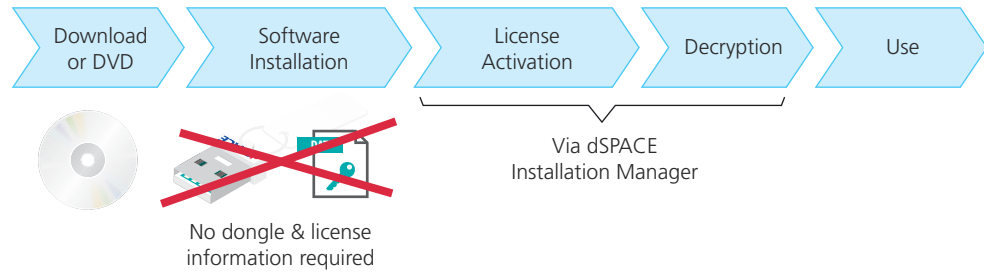
These are the main technology changes:

- The WibuKey and FlexNet technologies are replaced by the CodeMeter technology.
- Release-dependent **License.dsp** files are no longer used.
- The license information is Release-independent and stored in a CmContainer. A CmContainer can be a CmDongle or a CmActLicense (software license container).
- License activation, deactivation, and license updates are carried out via a database (dSPACE License Central) on the basis of tickets.



dSPACE License Central is hosted by dSPACE GmbH and is secured against unauthorized access by a Web gateway.

**New workflow**



As shown above, the software installation process is now license-independent. No license information and/or dongles are required during the installation. This simplifies the installation process significantly.

As a consequence, you can now install the complete dSPACE Release without any license. However, specific parts of dSPACE software, the documentation, demo models, .NET code, etc. are installed in encrypted archives on the end user's PC. To work with the installed dSPACE software, you have to activate licenses and then decrypt those parts you have licenses for.

**Tip**

License activation and decryption do not require administrator rights, so the end users can perform these steps themselves.

**Further information**

For detailed information on the CodeMeter licensing technology, refer to [Basics on CodeMeter Licensing Technology](#) ([Working with CodeMeter Licensing Technology](#)).

## New Setup Program for Software Installation

**Introduction**

In dSPACE Release 2017-B, the setup program is redesigned completely to simplify the installation process.

**Main features and benefits**

The new setup program, dSPACE Setup, provides the following main features and benefits:

- dSPACE Setup is the central setup program for all dSPACE software products and all installation tasks.
- It supports initial software installation, removing single product sets, removing a complete dSPACE Release, as well as modifying or repairing an existing installation.
- No license information and/or dongles are required during the installation.

- The installation is more transparent. dSPACE software is installed in larger units, called product sets. Each set contains all software components and options that are typically used together.

You can install and uninstall software only on the product set level. It is not possible to add or remove single products. Working with product sets is easy because they have descriptive names and provide a good balance between disk space requirements and usability.
- The installation is now performed without any user interaction during the installation process.

The new setup requests all information before the installation starts. You can force your host PC to shut down after the installation. This enables you to finish dSPACE software installation without having to attend.
- dSPACE Setup provides an automatic installation feature for unattended installations.

You can record an initial interactive installation and then use it to install dSPACE software on multiple host PCs automatically without user interaction.

Multiple installations

Note

You can install only one instance of a product set (for example, ControlDesk, AutomationDesk, Model Compare, ECU Interface Software, VEOS, TargetLink) on your host PC. Multiple installations of the same Release is not supported by the new setup. However, you can install different Releases of a product set on your host PC.

User interface

dSPACE Setup has a modern, clean user interface:

The screenshot shows the 'Modify' window for 'dSPACE Release 2017-B'. On the left, a progress bar indicates the current step is '2. PRODUCT SETS'. The main area is titled 'Product Sets' and contains a list of software components with checkboxes. A search bar at the top right of this section is empty. On the right side, there is a 'Product Set Description' for 'AutomationDesk', which includes a list of products like 'AutomationDesk Basic Version (AUD\_BASIC)' and 'AutomationDesk Server Basic Version (AUS\_BASIC)'. At the bottom, there are 'Configure', 'Abort', 'Back', and 'Next' buttons. A 'Required Disk Space' section at the bottom left shows '1.5 GB of free 144.8 GB'.

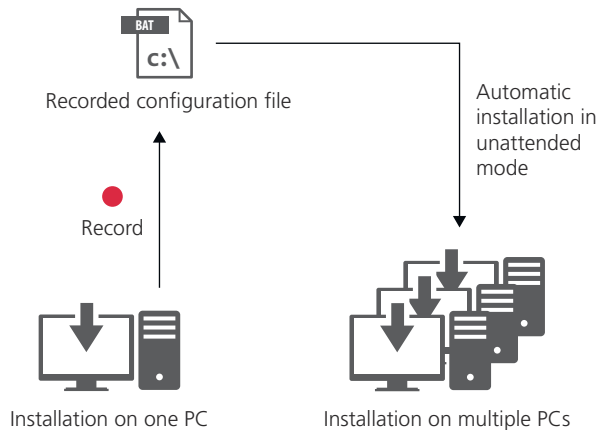
Checkbox display:

- Already installed —  AutomationDesk
- To be installed —  ConfigurationDesk - Implementatio...
- To be removed —  MotionDesk

As shown above, product selection becomes much easier because you can select product sets from a flat list. There is no longer a complex tree of installable software components you can or must select from, requiring intimate knowledge about which component is located where.

### Unattended installations

You can use the record mode of dSPACE Setup to record the installation configuration of dSPACE software and use the generated configuration file to install the software on other PCs automatically in the setup's unattended mode. This mode does not require any manual input.



### Further information

For more basics and detailed instructions, refer to [Basics on dSPACE Software Installation](#) ( [Installing dSPACE Software](#)).

## License Activation

### Introduction

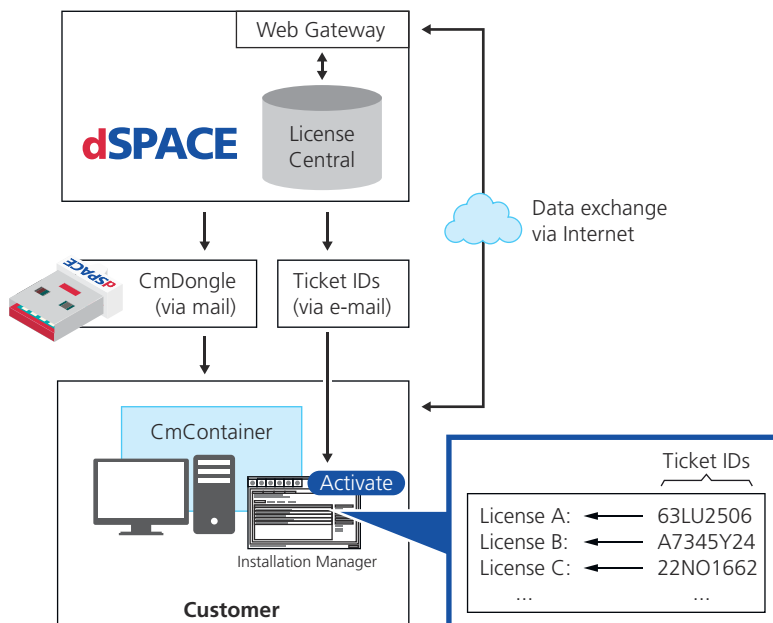
CodeMeter licenses are made available via CmContainers (CmDongles or CmActLicenses) on user PCs. After software installation, license activation via dSPACE Installation Manager is needed to run license-protected dSPACE software.

### Activation principle

License activation uses a ticket-based system. dSPACE GmbH hosts a license database: dSPACE License Central. Depending on your order, dSPACE specifies the attributes for each license and provides this information to the database. dSPACE License Central then generates a ticket ID for each license.

The ticket ID is a unique alphanumeric string of characters. The ticket ID of a license remains unchanged as long as the license exists, but the license information related to it can be updated, for example, if the maintenance period of a license is extended with an SMS order.

These ticket IDs are sent to you by e-mail. You have to enter the IDs in the Installation Manager and start the activation process. During this process, specific license activation files are transferred between dSPACE Installation Manager and dSPACE License Central.



You can activate licenses online, which requires an Internet connection between the Installation Manager and dSPACE License Central, or offline by means of file transfer via e-mail. dSPACE strongly recommends to use online activation whenever possible.

Online activation differs depending on how dSPACE Installation Manager can access the CmContainer.

In a final step after the activation process, encrypted parts of the software you have licenses for must be decrypted. You can then use the license-protected dSPACE software on your host PC.

#### Tip

License activation does not require administrator rights, so the end users can perform this step themselves.

#### Further information

For more basics and detailed instructions, refer to [License Activation \(Working with CodeMeter Licensing Technology\)](#).

## Decryption of Encrypted Archives of dSPACE Software

### Motivation

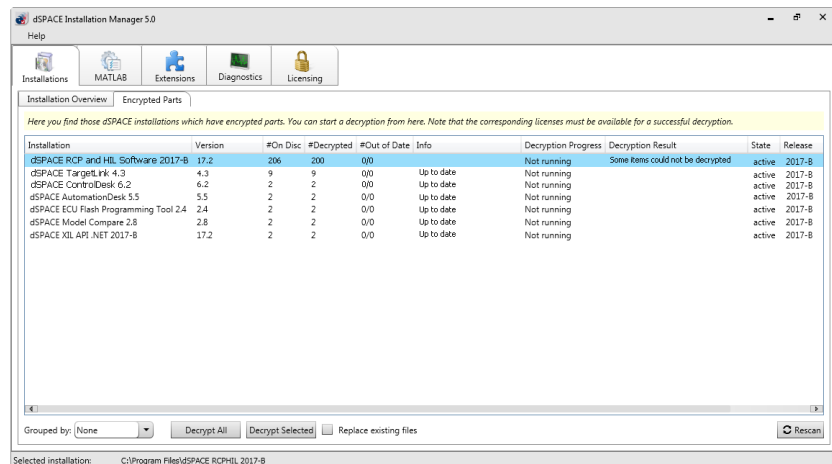
After software installation, specific parts of the dSPACE software, the documentation, demo models, .NET code are installed in license-protected and encrypted archives on the end user's PC. To work with the installed dSPACE software, you first have to activate licenses and then decrypt the encrypted parts of the installed products.

### Decryption process

You can decrypt only those parts of software products for which you have activated licenses. These licenses must be accessible from your host PC during the decryption process, for example, via a connected CmDongle.

During decryption, the relevant license is only checked and not blocked. Therefore, in a floating network scenario, no license is blocked by a license client on which a decryption process is performed.

Decryption is performed with dSPACE Installation Manager. dSPACE Installation Manager checks if encrypted parts are available and displays all installed software products/packages on your host PC that contain encrypted parts (see illustration below). Software products/packages containing none encrypted parts are not displayed.



You can run the decryption process either for selected software products or for all products that are displayed on the **Encrypted Parts** page.

Decryption is needed only once after software installation. Even if you install many product sets, decryption usually takes only a few minutes.

dSPACE Installation Manager also displays the progress of decryption and gives helpful status information, for example, if some parts cannot be decrypted.

#### Tip


Decryption does not require administrator rights, so the end users can perform this step themselves.



---

**Result** After initial decryption, all files remain decrypted on the host PC. However, whenever an update or patch is installed that contains encrypted components, you have to use dSPACE Installation Manager to decrypt them all again.

---

**Further information** For more basics and detailed instructions, refer to [Decrypting Encrypted Archives of dSPACE Software Installations](#) ( [Managing dSPACE Software Installations](#)).

## Compatibility Information

---

### Compatibility of dongles

- Green WibuKey dongles must be replaced and their License ID migrated to CmDongles. The CmDongles are automatically delivered with dSPACE Release 2017-B to owners of WibuKey dongles with licenses covered by a Software Maintenance Service (SMS) contract.
- CmDongles (Rev. 3-xxxx) delivered for dSPACE Release 2014-B up to and including dSPACE Release 2017-A are not prepared to use the CodeMeter licensing technology. They must be migrated to use dSPACE Release 2017-B and later.
- CmDongles shipped for dSPACE Release 2017-B and later support CodeMeter licensing without modifications.

#### Note

All CmDongles (Rev. 3-xxxx), including those shipped with dSPACE Release 2017-B and later, can still be used with dSPACE Releases 2017-A and earlier, because they can emulate WibuKey dongles.

---

### Parallel use of licensing technologies

The parallel use of old and new licensing technologies is possible.

You can use the WibuKey (or FlexNet) licensing technology required for products of earlier dSPACE Releases in parallel with the CodeMeter licensing technology on the same host PC.

For floating network licenses, the CodeMeter licensing technology cannot provide downward compatibility. If you need to use product versions of earlier dSPACE Releases, you have to keep your FlexNet License Server running in parallel. However, both FlexNet and CodeMeter floating network servers can be hosted on the same PC.

**Note**

The dSPACE EULA does not allow FlexNet and CodeMeter license servers to provide the *same floating network license* in parallel. If you have any questions or encounter any problems, contact dSPACE Support.

**Managing licenses for earlier dSPACE Releases**

To manage licenses for software installations from dSPACE Release 2017-A and earlier, you have to use the **dSPACE License Manager (Legacy)**. This is a separate tool that is always installed together with dSPACE Installation Manager 5.0 and later.

**dSPACE License Manager (Legacy)** provides the same license management functionalities as dSPACE Installation Manager 4.3 (distributed with dSPACE Release 2017-A) and earlier. This applies to dongle licenses (based on the WibuKey technology) as well as to floating network licenses (based on the FlexNet technology).

## Requirements for Communication

**Purpose**

To communicate with dSPACE License Central (via a web gateway), with CodeMeter tools from Wibu-Systems or other PCs in a LAN, your PC must be able to establish a TCP/IP connection.

**Communication with dSPACE License Central**

To access dSPACE License Central from your PC, you need an Internet connection. dSPACE Installation Manager connects to <https://licensing.dspace.de/gateways>, HTTPS, Port 443. dSPACE Installation Manager can work with proxy servers that you might use. It can also handle proxy servers with authentication (user name, password).

**Tip**

dSPACE Installation Manager uses the same settings for communication as Microsoft Internet Explorer. Therefore, you can change the setting in the Internet Explorer to solve any communication problems. In addition, you can check the Internet connection to dSPACE License Central by entering the following address in your web browser: <https://licensing.dspace.de>

Local firewalls and IT appliances must be configured accordingly to grant dSPACE Installation Manager access to the Internet.

**Internal LAN communication**

For LAN communication between CodeMeter software on different PCs, for example, for connections between floating network license servers and clients, a standard TCP/UDP protocol is required. The default port number is 22350, which

is an officially registered port number (IANA). If needed, you can change the port number.

For communication with CmContainers on remote PCs in a LAN, dSPACE Installation Manager uses the CodeMeter functionality. Thus, the connected PCs require the same open TCP/IP network port: 22350 (TCP and UDP).



# Aspects of Migrating from Previous Releases

---

## Introduction

After you install products of the current dSPACE Release, some additional steps might be necessary. The migration steps required when you update from the last dSPACE Release are described in the product-specific migration topics in this document. If you update from an earlier dSPACE Release, refer to the related *New Features and Migration* document.

## Migrating to dSPACE Release 2017-B

---

### Introduction

After you install Release 2017-B, some additional steps might be necessary.

---

### Migrating from dSPACE Release 2017-A

**Product-specific migration steps** Product-specific migration steps are usually performed automatically by the products. For exceptions, refer to the product-specific migration descriptions.

---

### Migrating from dSPACE Release 2016-B or earlier

To migrate from dSPACE Release 2016-B or earlier to Release 2017-B, you also have to perform the migration steps of the intervening dSPACE Releases. All of the required migration steps can be performed with Release 2017-B installed.

For more information on the required migration steps, refer to the *New Features and Migration* documents of the intervening dSPACE Releases.

---

### Previous release documents

The PDF files of previous Releases are called **NewFeaturesAndMigrationxx.pdf**, where xx stands for the Release number.

You can find the *New Features and Migration* files for previous Releases in the following locations:

- In the installation folder of the current dSPACE Help. Refer to `C:\Program Files\Common Files\dSPACE\Help 2017-B\Print\PreviousReleases`.
- On the dSPACE DVDs. Refer to `\Doc\PreviousReleases`.
- At [www.dspace.com/go/migration](http://www.dspace.com/go/migration) for download. Here, you can also find *New Features and Migration* documents for very early Releases.

# AutomationDesk

## Where to go from here

## Information in this section

New Features of AutomationDesk 5.5 .....	47
Migrating to AutomationDesk 5.5 .....	51

## New Features of AutomationDesk 5.5

## Where to go from here

## Information in this topic

New licensing for dSPACE products .....	47
General enhancements .....	48
Enhanced hardware support for electrical error simulation .....	49
Improved user documentation .....	49
Support of XIL API Framework .....	48
Usability improvements .....	48
Enhancements to the libraries .....	50
Custom Library .....	50
Main Library .....	50
Report library .....	50
XIL API Convenience library .....	50
XIL API library .....	50
Enhancements to the COM API .....	50

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained

files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

## General enhancements

### Usability improvements

- Enhanced handling of data objects within the Sequence Builder.  
You can enhance the block parameters of some automation blocks, by adding additional data objects to them. You can now add a data object to these automation blocks by using the mouse to drag the data object the Library Browser, the Project Manager, or the Library Favorites pane to the block in the Sequence Builder. In the Sequence Builder you can select a data object via mouse or keyboard and then edit it.
- Enhanced handling of the Signal Editor.  
The following enhancements are now available:
  - The **Comment** field allows you to edit multiline strings.
  - The **Open** dialog for STZ files memorizes the previously used folder.
  - STZ file contents can now be selected.
  - The STZ file linked with the TestCase is automatically saved when you save the AutomationDesk project.
  - Selected variables can be opened directly in the related mapping data object via the **Select in Mapping** command.
  - The handling of the custom columns was improved, e.g., you can rename the column names.
  - Support of fixed parameters with error handling if they are used in an incorrect context.
- Handling of fixed parameters.  
Fixed parameters can be changed only in the initialization phase of the simulation application. These parameters are marked in the Variable Browser as *Init Only*.  
If you specify fixed parameters for writing in the Mapping Editor or the Signal Editor, warnings are displayed.

**Support of XIL API Framework** AutomationDesk supports an XIL API server that you configure via an XIL API Framework.

To configure and use the XIL API Framework, the following features are added to AutomationDesk:

- New commands in the **Platforms** ribbon to configure, edit, initialize, and shutdown an XIL API framework.
- For editing the Framework configuration, the **Mapping Editor** has been completely redesigned. It now lets you specify the MAPorts and EESPorts to be used, the labels on testbench and framework side, and the mapping of these labels.
- In the XIL API library, the new blocks **SetValues**, **GetValues**, and **CheckValues** let you access variables from the framework configuration via their alias names.



- To use these blocks without a framework configuration, the framework context can be disabled by using the **Disable Framework Support** command. This adds the MAPort data object and changes the handling of the block's data objects. For more information on switching between the framework and the test bench context, refer to the descriptions in [Accessing Simulation Platforms](#).
- The XIL API Convenience demo provides an example of using an XIL API framework. In the **FrameworkConfiguration** demo folder, you find configuration files for SCALEXIO and VEOS as examples.

For more information, refer to [XIL API Framework](#) ([Accessing Simulation Platforms](#)).

**Enhanced hardware support for electrical error simulation** The EESPort in the XIL API and the XIL API Convenience libraries now supports DS5355/DS5390 High Current FIU systems.

**Improved user documentation** Together with the introduction of the new dSPACE Help, the structure of the user documentation for AutomationDesk was also changed from a document-oriented structure to a subject-oriented structure. The information on a specific subject is no longer spread across different documents, such as the AutomationDesk Guide, AutomationDesk Reference, etc. You can find the entire documentation for a specific subject, such as implementing signal-based tests or simulating electrical errors, under a single node in dSPACE Help.

The new documents in the subject-oriented structure are:

- [Introduction And Overview](#)
- [Basic Practices](#)
- [Implementing Signal-Based Tests](#)
- [Accessing Simulation Platforms](#)
- [Simulating Electrical Errors](#)
- [Accessing ControlDesk](#)
- [Accessing MotionDesk](#)
- [Accessing ModelDesk](#)
- [Accessing Real-Time Testing](#)
- [Accessing MATLAB](#)
- [Accessing Remote Calibration COM](#)
- [Accessing Remote Diagnostics COM](#)
- [Accessing RS232](#)
- [Accessing CANscope](#)
- [Accessing CANstress](#)
- [Automation](#)

The [Tutorial](#) is still available as a single document.

---

**Enhancements to the libraries**

The following libraries were enhanced:

**Main Library** The following data object is now available in the Main Library:

- **Bool**

This data object can be parameterized with `true` or `false`.

The following block of the Main Library was enhanced:

- **File**

The **Edit** dialog of the **File** data object allows you to configure a file as an attachment. Then, the specified file is added to the AutomationDesk project.


Relevant test data is then directly stored in the project and not saved to external files that are not included in a project's export, for example.

For more information, refer to [Main Library](#) ( [Basic Practices](#)).


**Report library** The Report library now offers a new automation block:

- **AddHTML**

This block lets you enter any valid HTML code to add custom-specific information to the generated HTML or PDF report.

For more information, refer to [Report](#) ( [Basic Practices](#)).

**XIL API library** The XIL API library now supports the XIL API framework concept, as previously described.

For more information, refer to [XIL API \(Model Access\)](#) ( [Accessing Simulation Platforms](#)).

**XIL API Convenience library** The XIL API Convenience library now supports the XIL API framework concept. The dialogs for parameterizing the MAPort and Variable data objects let you use the settings in the framework configuration.

**Custom Library** The library link mechanism for custom templates was changed. If you add a custom template to the Sequence Builder, it is represented by a **LibraryLink** block. This decouples the type of the custom template from its invoking block.

This lets you:

- Change the target of the **LibraryLink** block by editing the **Link** property in the **Properties** dialog.
- Use the dynamic link mode for *all* block types.
- Execute the custom template recursively and in **Parallel** blocks.

---

**Enhancements to the COM API**

The AutomationDesk COM API provides the following enhancements:

- Access to the new data object **Bool**, and the attachment of a **File** data object.
- Read access to the storage path of a linked custom library by using the **Path** method.
- Access to the XIL API framework configuration.

For more information, refer to  [Automation](#).

## Migrating to AutomationDesk 5.5


### General migration aspects


If you open an AutomationDesk project with a later AutomationDesk version, the software automatically detects whether migration is necessary. Click **OK** in the message dialog to start the migration. If you also want to continue working with the old project, you must not overwrite it with the migrated project, because the versions are not downward compatible. Save the migrated project to another path or name.

#### Note

Before you open an older project with the new AutomationDesk version, make sure that the following preconditions are fulfilled:

- You must create backups of the project and of the linked custom libraries.
- AutomationDesk must be running properly. The Log Viewer must not display any error messages.
- The built-in libraries, required custom libraries, and other packages must be loaded correctly.
- To import an older project to a new AutomationDesk version, the exported project or custom library must be available in ZIP format. The automatic migration does not support the XML format.

If you use a version control system, there are some preconditions for successful migration, refer to [How to Migrate Projects Under Version Control](#) ( [Basic Practices](#)).

For more information, refer to [Migrating AutomationDesk](#) ( [Introduction And Overview](#)).

### Custom Library

If you migrate your project to AutomationDesk 5.5, an instance of a custom template is internally replaced by a **LibraryLink** block.

Because the **LibraryLink** block creates an additional hierarchy level during execution time, the execution result might change in the following cases:

- You are using scripts in **Exec** or **ExecFile** blocks that use the `_AD_. _INFO_.Shift()` method with a numerical value.
- You are using the **RangeBlockDataObjects** block in a custom template with the **Shift** data object parameterized to shift the execution context outside the custom template's root level.

In these cases, you have to edit the shift parameters. In all other cases the migration of the custom templates does not require manual modifications.

There might be more changes to be noticed for instantiated custom templates, such as changed entries in an XML export, changed tool tips, or icons.



# Automotive Simulation Models (ASM)

## Where to go from here

## Information in this section

All ASM Blocksets .....	54
ASM Brake Hydraulics Blockset .....	55
ASM Diesel Engine Blockset .....	56
ASM Diesel Exhaust Blockset .....	62
ASM Diesel InCylinder Blockset .....	63
ASM Drivetrain Basic Blockset .....	64
ASM Environment Blockset .....	66
ASM Gasoline Engine Basic Blockset .....	67
ASM Gasoline Engine Blockset .....	68
ASM Gasoline InCylinder Blockset .....	72
ASM Traffic Blockset .....	73
ASM Trailer Blockset .....	75
ASM Truck Blockset .....	76
ASM Vehicle Dynamics Blockset .....	77

## Information in other sections

### [Migrating ASM Models \(📖 ASM User Guide\)](#)

Provides general information on the migration of ASM models.

# All ASM Blocksets

## Where to go from here

## Information in this section

<a href="#">New Features of All ASM Blocksets .....</a>	<a href="#">54</a>
<a href="#">Migrating All ASM Blocksets .....</a>	<a href="#">54</a>

## New Features of All ASM Blocksets

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

## Migrating All ASM Blocksets

### Migration strategy

The ASM migration supports the migration from the last ten Releases to dSPACE Release 2017-B. If you want to migrate from an older Release than the supported versions, the migration may fail. In this case, migrate to an intermediate Release and afterwards to the current Release. For more information, refer to [Migrating ASM Models \(📖 ASM User Guide\)](#).

# ASM Brake Hydraulics Blockset

## Migrating to ASM Brake Hydraulics Blockset 2.0.1

---

### **DESIRED\_BRAKE\_PRESSURE**

Memory has been added to the **DESIRED\_BRAKE\_PRESSURE** block to avoid an algebraic loop in the ASM Operator mode.

---

### Related topics

#### Basics

[Migrating ASM Models \(📖 ASM User Guide\)](#)

# ASM Diesel Engine Blockset

## Where to go from here

## Information in this section

New Features of ASM Diesel Engine Blockset 2.6 .....	56
Changes in the ASM Diesel Engine Demo Model .....	59
Migrating to ASM Diesel Engine Blockset 2.6 .....	60

## New Features of ASM Diesel Engine Blockset 2.6

### INTERCOOLER

The efficiency map that is based on the engine operating point (Map\_eta\_Cooler parameter) has been removed.

The Map\_eta\_Cooler\_phy parameter (efficiency map based on the physical model) has been renamed to Map\_eta\_Cooler.

The new Sw\_State\_Cooler parameter has been introduced to switch off the intercooler.

The p\_In\_InterCooler inport and p\_Out\_InterCooler outport have been removed, because the process in the model is isobaric.

The unit of the mass flow entering the block has been changed from [kg/s] to [kg/h].

A former version block (**INTERCOOLER\_5\_0**) has been created.

### EGRCOOLER

The efficiency map that is based on the engine operating point (Map\_eta\_Cooler parameter) has been removed.

The Map\_eta\_Cooler\_phy parameter (efficiency map based on the physical model) has been renamed to Map\_eta\_Cooler.

The new Sw\_State\_Cooler parameter has been introduced to switch off the EGR cooler.

The p\_In\_EGRCooler inport and p\_Out\_EGRCooler outport have been removed, because the process in the model is isobaric.

The unit of the mass flow entering the block has been changed from [kg/s] to [kg/h].

A former version block (**EGRCOOLER\_4\_0**) has been created.



---

**EGR\_VALVE**

The former **EGR\_VALVE** block has been split into a physical model (**EGR\_VALVE** in dSPACE Release 2017-B) and a mechanical model (**EGR\_VALVE\_MECHANICAL**).

The new **EGR\_VALVE** block is vectorial, i.e., it can simulate multiple branches of the air path simultaneously.

The mass flows of the air and the exhaust gas are calculated on the basis of the mass fractions. The direction of the mass flow is switched on the basis of the pressure difference on both sides of the valve. The T\_Out\_EGR[degC] output has been removed, because the process in the model is isothermal.

A former version block (**EGR\_VALVE\_6\_0**) has been created.

---

**EGR\_VALVE\_MECHANICAL**

The new **EGR\_VALVE\_MECHANICAL** block calculates the position of the EGR valve according to a control signal. By switching off the supply of the valve, a constant position (rest position) is specified.

---

**INTAKE\_MANIFOLD**

In this version, the sum blocks of the incoming mass flows and the enthalpy flows inside the energy balance are vectorial, i.e., the **INTAKE\_MANIFOLD** block can simulate multiple branches of the air path simultaneously.

The block considers the mass flow of the fuel, which can come from fuel evaporation system.

The ratio\_m\_Exh\_InMan[0\_1] output has been removed. Use the Xsi\_Exh\_InMan[0\_1] output instead.

A former version block (**INTAKE\_MANIFOLD\_8\_0**) has been created.

---

**EXHAUST\_MANIFOLD**

In this version, the sum blocks of the incoming mass flows and the enthalpy flows inside the energy balance are vectorial, i.e., the **EXHAUST\_MANIFOLD** block can simulate multiple branches of the air path simultaneously.

The block differentiates between the air and the exhaust gas in the calculation of the temperature and the pressure.

It has new outputs for the gas constant (R\_ExhMan[J](kgK)) as well as the mass fractions of the air (Xsi\_Air\_ExhMan[0\_1]) and the exhaust gas (Xsi\_Exh\_ExhMan[0\_1]) inside the exhaust manifold.

A former version block (**EXHAUST\_MANIFOLD\_10\_0**) has been created.

---

**LP\_EGRCOOLER**

The Map\_eta\_Cooler\_phy parameter has been renamed to Map\_eta\_Cooler.

The new Sw\_State\_Cooler parameter has been introduced to switch off the intercooler.

The Sw\_LP\_EGRCooler\_On parameter has been removed.

The unit of the mass flow entering the block has been changed from [kg/s] to [kg/h].

A former version block (**LP\_EGRCOOLER\_4\_0**) has been created.

---

#### **LP\_EGR\_VALVE**

The former **LP\_EGR\_VALVE** block has been split into a physical model (**LP\_EGR\_VALVE** in dSPACE Release 2017-B) and a mechanical model (**LP\_EGR\_VALVE\_MECHANICAL**). The new **LP\_EGR\_VALVE** block is vectorial, i.e., the block can simulate multiple branches of the air path simultaneously.

The mass flows of the air and the exhaust gas are calculated on the basis of the mass fractions.

The direction of the mass flow is switched on the basis of the pressure difference on both sides of the valve.

The T\_Out\_LPEGR[degC] outport has been removed, because the process in the model is isothermal.

A former version block (**LP\_EGR\_VALVE\_6\_0**) has been created.

---

#### **LP\_EGR\_VALVE\_MECHANICAL**

The new **LP\_EGR\_VALVE\_MECHANICAL** block calculates the position of the low-pressure EGR valve according to a control signal. By switching off the supply of the valve, a constant position (rest position) is specified.

---

#### **EXHAUSTTHROTTLE**

The former **EXHAUSTTHROTTLE** block has been split into a physical model (**EXHAUSTTHROTTLE** in dSPACE Release 2017-B) and a mechanical model (**EXHAUSTTHROTTLE\_MECHANICAL**).

A former version block (**EXHAUSTTHROTTLE\_5\_0**) has been created.

---

#### **EXHAUSTTHROTTLE\_MECHANICAL**

A new **EXHAUSTTHROTTLE\_MECHANICAL** block calculates the position of the exhaust throttle according to a control signal. By switching off the supply of the valve, a constant position (rest position) is specified.

---

#### **LP\_EXHAUST\_MANIFOLD**

The **LP\_EXHAUST\_MANIFOLD** block differentiates between the air and the exhaust gas in the calculation of the temperature and the pressure.

The block has a new outport for the gas constant ( $R_{LPExhMan}[J/(kgK)]$ ) as well as the mass fractions of the air ( $X_{si\_Air\_LPExhMan}[0\_1]$ ) and the exhaust gas ( $X_{si\_Exh\_LPExhMan}[0\_1]$ ) of the manifold.

The unit of the mass flows entering the block has been changed from [kg/h] to [kg/s].

The unit for the temperature entering or leaving the block has been changed from [Kelvin] to [°C].

A former version block (**LP\_EXHAUST\_MANIFOLD\_1\_0**) has been created.

---

<b>LP_INTAKE_MANIFOLD</b>	<p>In this version, the sum blocks of the enthalpy flows inside the energy balance are vectorial, i.e., the <b>LP_INTAKE_MANIFOLD</b> block can simulate multiple branches of the air path simultaneously.</p> <p>There is a new outport for the gas constant of the manifold (R_LPMan[J/(kgK)]).</p> <p>The T_In_Comp[degC] inport has been renamed to T_Out_LPInMan[degC], which is the temperature downstream of the low-pressure intake manifold. The signal is used to simulate the back flow.</p> <p>A former version block (<b>LP_INTAKE_MANIFOLD_6_0</b>) has been created.</p>
<b>THROTTLE_MECHANICAL</b>	<p>The <b>THROTTLE_MECHANICAL</b> block has a new Sw_State_ValveMechanical parameter (mask value Sw_State_Throttle) to switch off the model.</p>

---

## Changes in the ASM Diesel Engine Demo Model

---

<b>AirPath</b>	<p>The tags of the <b>Goto/From</b> blocks in the <b>AirPath</b> are not specific to a certain engine type anymore. The tags are renamed from TAG_EngineDiesel_* to TAG_Engine_*.</p> <p>The <b>GotoTagVisibility</b> blocks have to be used.</p>
<b>Engine</b>	<p>The <b>EngineOperationPoint</b> system has been added to get the operating point of the engine in the ASMSignalBus.</p>
<b>Environment</b>	<p>The ambient conditions pressure and temperature have been added to the <b>Road</b> subsystem of the <b>Environment</b> system. The switch for replacing the conditions with the measured data and the new <b>AMBIENT</b> block have been added. The switch has been moved from <b>Environment/Maneuver</b> to <b>Environment/Road</b>. In the <b>Road</b> subsystem, the ambient conditions have been added to the ASMSignalBus.</p>
<b>FuelConsumption</b>	<p>The <b>FuelConsumption</b> system (<b>ASM_EngineDiesel / MDLUserInterface / EngineDiesel / MDL_DISP / FuelConsumption</b>) has been modified.</p> <p>The fuel consumption is summarized in one system with outputs be[gjkWh] (brake specific fuel consumption), FuelConsumption[L 100km], FuelConsumptionAver[L 100km] and FuelConsumption[L h].</p> <p>The FuelConsumptionAver[L 100km] signal is the fuel consumption averaged over the driving distance.</p>

There is a new FuelConsumption signal. If the velocity of the vehicle is below a threshold value (default value is 1 km/h), the fuel consumption is given in [L/h], otherwise in [L/100 km].

## Migrating to ASM Diesel Engine Blockset 2.6

---

<b>INTERCOOLER</b>	The library link of the block has been changed to <b>FormerVersion/INTERCOOLER_5_0</b> .
--------------------	--

---

<b>INTAKE_MANIFOLD</b>	The library link of the block has been changed to <b>FormerVersion/INTAKE_MANIFOLD_8_0</b> .
------------------------	--

---

<b>EXHAUST_MANIFOLD</b>	The library link of the block has been changed to <b>FormerVersion/EXHAUST_MANIFOLD_10_0</b> .
-------------------------	--

---

<b>EGR_VALVE</b>	The library link of the block has been changed to <b>FormerVersion/EGR_VALVE_6_0</b> .
------------------	--

---

<b>EGRCOOLER</b>	The library link of the block has been changed to <b>FormerVersion/EGRCOOLER_4_0</b> .
------------------	--

---

<b>LP_INTAKE_MANIFOLD</b>	The library link of the block has been changed to <b>FormerVersion/LP_INTAKE_MANIFOLD_6_0</b> .
---------------------------	---

---

<b>LP_EXHAUST_MANIFOLD</b>	The library link of the block has been changed to <b>FormerVersion/LP_EXHAUST_MANIFOLD_1_0</b> .
----------------------------	--

---

<b>LP_EGR_VALVE</b>	The library link of the block has been changed to <b>FormerVersion/LP_EGR_VALVE_6_0</b> .
---------------------	---

---

<b>LP_EGRCOOLER</b>	The library link of the block has been changed to <b>FormerVersion/LP_EGRCOOLER_4_0</b> .
---------------------	---

---

<b>EXHAUSTTHROTTLE</b>	The library link of the block has been changed to <b>FormerVersion/EXHAUSTTHROTTLE_5_0</b> .
------------------------	--

---

---

<b>SWITCHES_INTERCOOLER</b>	The block has been discontinued. The library link of the block has been changed to <b>FormerVersion/SWITCHES_INTERCOOLER_2_0</b> .
<b>SWITCHES_EGR_COOLER</b>	The block has been discontinued. The library link of the block has been changed to <b>FormerVersion/SWITCHES_EGR_COOLER_2_0</b> .
<b>SWITCHES_EGR_VALVE</b>	The block has been discontinued. The library link of the block has been changed to <b>FormerVersion/SWITCHES_EGR_VALVE_2_0</b> .
<b>THROTTLE_VALVE</b>	Signals in the ASMSignalBus have been renamed from mdot_Out_Throttle[kg h] to mdot_Throttle[kg h] and T_Out_Throttle[degC] to T_Throttle[degC]. During the migration, the signal names are changed to the original names.
<b>THROTTLE_MECHANICAL</b>	<p>The name of the bus of the <b>THROTTLE_MECHANICAL</b> component in the ASMSignalBus has been renamed from Throttle_Mechanical to ThrottleMechanical. During the migration, the original name is restored.</p> <p>The value of the new Sw_State_ValveMechanical parameter (mask value Sw_State_Throttle) is set to 1, i.e., the valve is open.</p>

---

**Related topics****Basics**

[Migrating ASM Models \(📖 ASM User Guide\)](#)

# ASM Diesel Exhaust Blockset

## Where to go from here

## Information in this section

New Features of ASM Diesel Exhaust Blockset 2.1.5 .....	62
Changes in the ASM Diesel Exhaust Demo Model .....	62
Migrating to ASM Diesel Exhaust Blockset 2.1.5 .....	62

## New Features of ASM Diesel Exhaust Blockset 2.1.5

### DIESEL\_PARTICULATE\_FILTER

The **DIESEL\_PARTICULATE\_FILTER** block has the new lambda\_Out\_DPF[] outputport.

## Changes in the ASM Diesel Exhaust Demo Model

### ExhaustSystem

The demos of the exhaust system (**ExhaustSystem\_DOC\_DPF**, **ExhaustSystem\_DOC\_DPF\_SCR**, **ExhaustSystem\_DOC\_DPF\_NonAir\_SCR**) have the new lambda\_Out\_DPF[] outputport.

## Migrating to ASM Diesel Exhaust Blockset 2.1.5

### DIESEL\_PARTICULATE\_FILTER

The new outputport is terminated.

## Related topics

### Basics

[Migrating ASM Models \(📖 ASM User Guide\)](#)

# ASM Diesel InCylinder Blockset

## Changes in the ASM Diesel InCylinder Demo Model

---

### **Environment**

The ambient conditions pressure and temperature have been added to the **Road** subsystem of the **Environment** system. The switch for replacing the conditions with the measured data and the new **AMBIENT** block have been added. The switch has been moved from **Environment/Maneuver** to **Environment/Road**. In the **Road** subsystem, the ambient conditions have been added to the ASMSignalBus.

## ASM Drivetrain Basic Blockset

### Where to go from here

### Information in this section

New Features of ASM Drivetrain Basic Blockset 5.0 .....	64
Migrating to ASM Drivetrain Basic Blockset 5.0 .....	65

## New Features of ASM Drivetrain Basic Blockset 5.0

### AMBIENT

The new **AMBIENT** block calculates the ambient temperature and pressure depending on the altitude. The block is part of the **Environment/Road** system.

### Dual-clutch transmission demo

The library now includes a new demo to simulate a dual-clutch transmission. The demo consists of complex transmission models with the related soft ECU and simplified the models for the engine, vehicle, and environment.

Detailed models for shafts, synchronizers and shift elements are used to build the transmission mechanics. There is also a user interface to configure the mechanical structure of the gearbox. You can specify the number of gear selectors, assign respective gears, and configure the general gearbox structure.

The gear selectors and the clutches are hydraulically actuated. The hydraulic system can be divided into hydraulic supply, clutch, and gearshift actuation.

To cover a comprehensive simulation of the dual-clutch transmission, several new blocks are introduced to the library. These blocks are:

- **CHAMBER**
- **COMMON\_HYDRAULICS\_PARAMETERS**
- **DCT\_GEARBOX\_MAPPING**
- **DCT\_GEARBOX\_TOPOLOGY**
- **DIRECTIONAL\_4\_2\_VALVE**
- **DOUBLE\_ACTING\_CYLINDER**
- **INPUT\_SHAFT\_SPEED**
- **PRESSURE\_ACTUATED\_CLUTCH**
- **PRESSURE\_CONTROL\_3\_2\_VALVE**
- **PRESSURE\_RELIEF\_VALVE**
- **PUMP**
- **SIMPLE\_GEAR**
- **SINGLE\_ACTING\_CYLINDER**
- **SYNCHRONIZER**



- DCT\_PRESELECTED\_GEAR
- DCT\_SETUP
- DCT\_STATE
- DIRECTIONAL\_VALVE\_CONTROL
- DUAL\_CLUTCH\_ENGINE\_INTERVENTION
- DUAL\_CLUTCH\_PRESSURE\_CONTROL
- DUAL\_CLUTCH\_PRESSURE\_SET
- ENGINE\_TORQUE\_SET\_INTERVENTION
- GEAR\_SELECTOR\_POSITION\_CONTROL
- GEAR\_SELECTOR\_POSITION\_SET
- LINE\_PRESSURE\_CONTROL
- PRESSURE\_VALVE\_CONTR

For more information on the demo, refer to [ASM Drivetrain Basic Model Description](#).

---

### Rearranged library

The library blocks have been rearranged due to the new introduced dual-clutch transmission demo. The blocks in the library are grouped according to their functionality. Moreover, several new blocks have been introduced to cover a comprehensive simulation of the dual-clutch transmission.

## Migrating to ASM Drivetrain Basic Blockset 5.0

---

### Relocated library blocks

Blocks that have been moved in the library are automatically migrated. During the migration, the links to these blocks are changed according to the new positions in the library.

---

### Related topics

#### Basics

[Migrating ASM Models \(ASM User Guide\)](#)

# ASM Environment Blockset

## New Features of ASM Environment Blockset 4.7

---

**ROAD** The **ROAD** block can calculate trajectories for the ASM vehicle and traffic fellows that are based on ModelDesk trajectory shapes.

---

**MANEUVER\_SCHEDULER** The maneuver scheduler has been improved so it is possible to set the steering torque stimulus in addition the steering wheel angle.

---

**Related topics**

**Basics**

<a href="#">New Features of ModelDesk 4.6 .....</a>	<a href="#">137</a>
---	---------------------

# ASM Gasoline Engine Basic Blockset

## Changes in the ASM Engine Gasoline Basic Demo Model

---

<b>Engine</b>	The <b>EngineOperationPoint</b> system has been added to get the operating point of the engine in the ASMSignalBus.
<b>FuelConsumption</b>	<p>The <b>FuelConsumption</b> system (<b>ASM_EngineGasolineBasic / MDLUserInterface / EngineGasBas / MDL_DISP / FuelConsumption</b>) has been modified.</p> <p>The fuel consumption is summarized in one system with outputs be[g kWh] (brake specific fuel consumption), FuelConsumption[L 100km], FuelConsumptionAver[L 100km] and FuelConsumption[L h].</p> <p>The FuelConsumptionAver[L 100km] signal is the fuel consumption averaged over the driving distance.</p> <p>There is a new FuelConsumption signal. If the velocity of the vehicle is below a threshold value (default value is 1 km/h), the fuel consumption is given in [L/h], otherwise in [L/100 km].</p>

# ASM Gasoline Engine Blockset

## Where to go from here

## Information in this section

New Features of ASM Gasoline Engine Blockset 4.0 .....	68
Changes in the ASM Engine Gasoline Demo Model .....	70
Migrating to ASM Gasoline Engine Blockset 4.0 .....	70

## New Features of ASM Gasoline Engine Blockset 4.0

### INTERCOOLER

The efficiency map that is based on the engine operating point (Map\_eta\_Cooler parameter) has been removed.

The Map\_eta\_Cooler\_phy parameter (efficiency map based on the physical model) has been renamed to Map\_eta\_Cooler.

The new Sw\_State\_Cooler parameter has been introduced to switch off the intercooler.

The p\_In\_InterCooler inport and p\_Out\_InterCooler outport have been removed, because the process in the model is isobaric.

The unit of the mass flow entering the block has been changed from [kg/s] to [kg/h].

A former version block (**INTERCOOLER\_5\_0**) has been created.

### EGRCOOLER

The efficiency map that is based on the engine operating point (Map\_eta\_Cooler parameter) has been removed.

The Map\_eta\_Cooler\_phy parameter (efficiency map based on the physical model) has been renamed to Map\_eta\_Cooler.

The new Sw\_State\_Cooler parameter has been introduced to switch off the EGR cooler.

The p\_In\_EGRCooler inport and p\_Out\_EGRCooler outport have been removed, because the process in the model is isobaric.

The unit of the mass flow entering the block has been changed from [kg/s] to [kg/h].

A former version block (**EGRCOOLER\_4\_0**) has been created.

---

**EGR\_VALVE**

The former **EGR\_VALVE** block has been split into a physical model (**EGR\_VALVE** in dSPACE Release 2017-B) and a mechanical model (**EGR\_VALVE\_MECHANICAL**).

The new **EGR\_VALVE** block is vectorial, i.e., it can simulate multiple branches of the air path simultaneously.

The mass flows of the air and the exhaust gas are calculated on the basis of the mass fractions. The direction of the mass flow is switched on the basis of the pressure difference on both sides of the valve. The T\_Out\_EGR[degC] output has been removed, because the process in the model is isothermal.

A former version block (**EGR\_VALVE\_5\_0**) has been created.

---

**EGR\_VALVE\_MECHANICAL**

The new **EGR\_VALVE\_MECHANICAL** block calculates the position of the EGR valve according to a control signal. By switching off the supply of the valve, a constant position (rest position) is specified.

---

**INTAKE\_MANIFOLD**

In this version, the sum blocks of the incoming mass flows and the enthalpy flows inside the energy balance are vectorial, i.e., the **INTAKE\_MANIFOLD** block can simulate multiple branches of the air path simultaneously.

The block considers the mass flow of the fuel, which can come from fuel evaporation system.

The ratio\_m\_Exh\_InMan[0\_1] output has been removed. Use the Xsi\_Exh\_InMan[0\_1] output instead.

A former version block (**INTAKE\_MANIFOLD\_6\_0**) has been created.

---

**EXHAUST\_MANIFOLD**

In this version the sum blocks of the incoming mass flows and the enthalpy flows inside the energy balance are vectorial, i.e. the **EXHAUST\_MANIFOLD** block can simulate several benches of the air path simultaneously.

The block differentiates between the air and the exhaust gas in the calculation of the temperature and the pressure.

It has new outputs for the gas constant (R\_ExhMan[J](kgK)) as well as the mass fractions of the air (Xsi\_Air\_ExhMan[0\_1]) and the exhaust gas (Xsi\_Exh\_ExhMan[0\_1]) inside the exhaust manifold.

A former version block (**EXHAUST\_MANIFOLD\_7\_0**) has been created.

---

**THROTTLE\_MECHANICAL**

The **THROTTLE\_MECHANICAL** block has a new Sw\_State\_ValveMechanical parameter (mask value Sw\_State\_Throttle) to switch off the model.

---

## Changes in the ASM Engine Gasoline Demo Model

<b>AirPath</b>	<p>The tags of the <b>Goto/From</b> blocks in the <b>AirPath</b> are not specific for a certain engine type anymore. The tags are renamed from TAG_EngineGasoline_* to TAG_Engine_*.</p> <p>The <b>GotoTagVisibility</b> blocks have to be used.</p>
<b>Engine</b>	<p>The <b>EngineOperationPoint</b> system has been added to get the operating point of the engine in the ASMSignalBus.</p>
<b>Environment</b>	<p>The ambient conditions pressure and temperature have been added to the <b>Road</b> subsystem of the <b>Environment</b> system. The switch for replacing the conditions with the measured data and the new <b>AMBIENT</b> block have been added. The switch has been moved from <b>Environment/Maneuver</b> to <b>Environment/Road</b>. In the <b>Road</b> subsystem, the ambient conditions have been added to the ASMSignalBus.</p>
<b>FuelConsumption</b>	<p>The <b>FuelConsumption</b> system (<b>ASM_EngineGasoline / MDLUserInterface / EngineGasoline / MDL_DISP / FuelConsumption</b>) has been modified.</p> <p>The fuel consumption is summarized in one system with outputs be[g kWh] (brake specific fuel consumption), FuelConsumption[L 100km], FuelConsumptionAver[L 100km] and FuelConsumption[L h].</p> <p>The FuelConsumptionAver[L 100km] signal is the fuel consumption averaged over the driving distance.</p> <p>There is a new FuelConsumption signal. If the velocity of the vehicle is below a threshold value (default value is 1 km/h), the fuel consumption is given in [L/h], otherwise in [L/100 km].</p>

## Migrating to ASM Gasoline Engine Blockset 4.0

<b>INTERCOOLER</b>	<p>The library link of the block has been changed to <b>FormerVersion/INTERCOOLER_5_0</b>.</p>
<b>INTAKE_MANIFOLD</b>	<p>The library link of the block has been changed to <b>FormerVersion/INTAKE_MANIFOLD_6_0</b>.</p>

<b>EXHAUST_MANIFOLD</b>	The library link of the block has been changed to <b>FormerVersion/EXHAUST_MANIFOLD_7_0</b> .
<b>EGR_VALVE</b>	The library link of the block has been changed to <b>FormerVersion/EGR_VALVE_5_0</b> .
<b>EGRCOOLER</b>	The library link of the block has been changed to <b>FormerVersion/EGRCOOLER_4_0</b> .
<b>SWITCHES_INTERCOOLER</b>	The block has been discontinued. The library link of the block has been changed to <b>FormerVersion/SWITCHES_INTERCOOLER_1_0</b> .
<b>SWITCHES_EGR_COOLER</b>	The block has been discontinued. The library link of the block has been changed to <b>FormerVersion/SWITCHES_EGR_COOLER_1_0</b> .
<b>SWITCHES_EGR_VALVE</b>	The block has been discontinued. The library link of the block has been changed to <b>FormerVersion/SWITCHES_EGR_VALVE_1_0</b> .
<b>THROTTLE_VALVE</b>	The following signals in the ASMSignalBus have been renamed from mdot_Out_Throttle[kg h] to mdot_Throttle[kg h] and T_Out_Throttle[degC] to T_Throttle[degC]. During the migration, the signal names are changed to the original names.
<b>THROTTLE_MECHANICAL</b>	The name of the bus of the <b>THROTTLE_MECHANICAL</b> component in the ASMSignalBus has been renamed from Throttle_Mechanical to ThrottleMechanical. During the migration, the original name is restored. The value of the new Sw_State_ValveMechanical parameter (mask value Sw_State_Throttle) is set to 1, i.e., the valve is open.

**Related topics****Basics**

[Migrating ASM Models \(📖 ASM User Guide\)](#)

## ASM Gasoline InCylinder Blockset

### Changes in the ASM Gasoline InCylinder Demo Model

---

#### **Environment**

The ambient conditions pressure and temperature have been added to the **Road** subsystem of the **Environment** system. The switch for replacing the conditions with the measured data and the new **AMBIENT** block have been added. The switch has been moved from **Environment/Maneuver** to **Environment/Road**. In the **Road** subsystem, the ambient conditions have been added to the ASMSignalBus.



# ASM Traffic Blockset

## Where to go from here

## Information in this section

New Features of ASM Traffic Blockset 3.7 .....	73
Changes in the ASM Traffic Demo Model .....	73
Migrating to ASM Traffic Blockset 3.7 .....	74

## New Features of ASM Traffic Blockset 3.7

### TRAFFIC\_SIGN\_SENSOR\_DYNAMIC\_CALCULATION

The **TRAFFIC\_SIGN\_SENSOR\_DYNAMIC\_CALCULATION** block is a new block for detecting the new traffic sign properties of traffic objects. The block can also handle objects with changing object states (e.g., traffic lights) and changing properties due to changing object states (e.g., variable message signs). The output dimensions can be parameterized. The block is compatible with the existing blocks for the traffic sign sensor parameterization.

### OBJECT\_SENSOR\_2D\_CALCULATION

The sensor is now able to detect continuous shapes of a road network. Continuous shapes can be specified and assigned to a road network by using the ModelDesk Road Generator.

## Related topics

### Basics

New Features of ModelDesk 4.6 .....	137
-------------------------------------	-----

## Changes in the ASM Traffic Demo Model

### Traffic\_Sign\_Sensor

The traffic sign sensor (**TRAFFIC\_SIGN\_SENSOR\_CALCULATION**) has been replaced by the new traffic sign sensor (**TRAFFIC\_SIGN\_SENSOR\_DYNAMIC\_CALCULATION**) for the new traffic sign properties.

---

**Objects\_DataStreams**                      The **MDUserInterface** subsystem (**Environment/MDL\_PAR**) has been extended by a new block to set object states for objects with multiple states, e.g., traffic lights or variable message signs.

---

**MD\_Instrumentation**                      The **MD\_Instrumentation** block in the **MotionDesk Interface** subsystem has been extended by data streams to set the states for objects with multiple states, e.g., traffic lights or variable message signs.

---

## Migrating to ASM Traffic Blockset 3.7

---

**OBJECT\_SENSOR\_2D\_CALCULATION**                      The new DiscreteObject\_Data[] inport is connected with [100, 0 0].

---

**OBJECT\_SENSOR\_2D\_PARAMETERS**                      The new Object\_Detection\_Mode outputport is terminated.  
The new Sw\_Object\_Detection\_Mode parameter is set to 1.

---

**OBJECT\_SENSOR\_2D\_GEOMETRY\_PARAMETERS**                      The new Object\_Detection\_Mode outputport is terminated.  
The new Sw\_Object\_Detection\_Mode parameter is set to 1.

---

### Related topics

#### Basics

[Migrating ASM Models \(📖 ASM User Guide\)](#)

# ASM Trailer Blockset

## Where to go from here

## Information in this section

Changes in the ASM Trailer Demo Model .....	75
Migrating to ASM Trailer Blockset 2.6.2 .....	75

## Changes in the ASM Trailer Demo Model

### Objects\_DataStreams

The **MDLUserInterface (Environment/MDL\_PAR)** has been extended by a new block to set object states for objects with multiple states, e.g., traffic lights or variable message signs.

### MD\_Instrumentation

The **MD\_Instrumentation** block in the MotionDesk Interface has been extended by data streams to set the states for objects with multiple states, e.g., traffic lights or variable message signs.

## Migrating to ASM Trailer Blockset 2.6.2

### TIRE\_MODEL\_TMEASY\_ TRAILER\_\*\*\*

The calculation in the blocks has been corrected: The bore torque is reduced to zero during forward movement.

## Related topics

### Basics

[Migrating ASM Models \(📖 ASM User Guide\)](#)

# ASM Truck Blockset

## Where to go from here

## Information in this section

<a href="#">Changes in the ASM Truck Demo Model .....</a>	<a href="#">76</a>
<a href="#">Migrating to ASM Truck Blockset 3.0.2 .....</a>	<a href="#">76</a>

## Changes in the ASM Truck Demo Model

### Objects\_DataStreams

The **MDLUserInterface (Environment/MDL\_PAR)** has been extended by a new block to set object states for objects with multiple states, e.g., traffic lights or variable message signs.

### MD\_Instrumentation

The **MD\_Instrumentation** block in the MotionDesk Interface has been extended by data streams to set the states for objects with multiple states, e.g., traffic lights or variable message signs.

## Migrating to ASM Truck Blockset 3.0.2

### TIRE\_MODEL\_TMEASY\_\*\*\*

The calculation in the blocks has been corrected: The bore torque is reduced to zero during forward movement.

## Related topics

### Basics

[Migrating ASM Models \(📖 ASM User Guide\)](#)


# ASM Vehicle Dynamics Blockset

**Where to go from here**

**Information in this section**

New Features of ASM Vehicle Dynamics Blockset 4.0 .....	77
Changes in the ASM Vehicle Dynamics Demo Model .....	77
Migrating to ASM Vehicle Dynamics Blockset 4.0 .....	77

## New Features of ASM Vehicle Dynamics Blockset 4.0

**TIRE MF** A new implementation of Magic Formula according to Magic Formula 6.1 has been added to the blockset. For more information on the model, refer to [MagicFormula Tire Model 6.1](#) ( [ASM Vehicle Dynamics Addendum](#)).

**TIRE MODE** The blockset contains the new **TIRE\_MODE** block that is used to mirror the calculation of Magic Formula.

## Changes in the ASM Vehicle Dynamics Demo Model

**Objects\_DataStreams** The **MDLUserInterface** (**Environment/MDL\_PAR**) has been extended by a new block to set object states for objects with multiple states, e.g., traffic lights or variable message signs.

**MD\_Instrumentation** The **MD\_Instrumentation** block in the MotionDesk Interface has been extended by data streams to set the states for objects with multiple states, e.g., traffic lights or variable message signs.

## Migrating to ASM Vehicle Dynamics Blockset 4.0

**TIRE\_MODEL\_TMEASY\_\*\*\*** The calculation in the blocks has been corrected: The bore torque is reduced to zero during forward movement.

---

**STEERING\_3DOF\_  
VARIABLE\_RATIO**

The new SW\_DriverArm parameter has been added.

The reset logic for System states has been updated to facilitate plausible switching from steering mode 3 to 2.

The new TrqSensor\_Steering[Nm] signal has been added to switch the sensor torque based on EPS motor location.

A signal specification has been added to avoid Simulink signal dimension error.

---

**SOFT\_ECU\_  
POWERSTEERING**

The TrqSpring\_SteeringColumn[Nm] inport has been renamed to TrqSensor\_Steering[Nm].

---

**Related topics**

**Basics**

[Migrating ASM Models \(📖 ASM User Guide\)](#)

# Bus Manager (Stand-Alone)

## Where to go from here

## Information in this section

Features of the Bus Manager (Stand-Alone) 6.0 .....	79
Migrating to Bus Manager (Stand-Alone) 6.0 .....	81

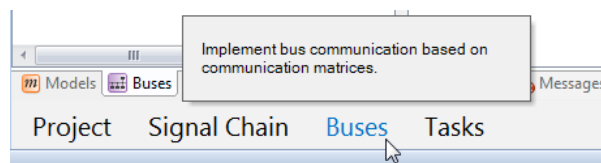
## Features of the Bus Manager (Stand-Alone) 6.0

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### User interface consists of view sets for specific purposes

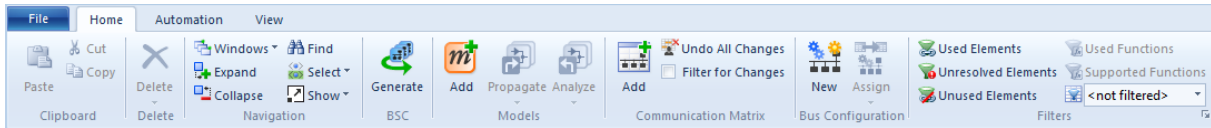
The user interface of the Bus Manager (stand-alone) now offers different view sets for specific purposes. You can switch between view sets by using the navigation bar:



The available panes of each view set serve a specific purpose. For example, the **Project** view set contains the **Project Manager** to perform project and application management tasks.

For each view set, the **Home** ribbon contains specific commands suitable for the purpose of the view set.

The following illustration shows the view set-specific **Home** ribbon of the **Buses** view set:



If the available view sets do not meet your requirements, you can customize them or create additional view sets with the panes of your choice.

For more information, refer to [Elements of the Bus Manager \(Stand-Alone\) Implementation Guide](#).

### AUTOSAR 4.3.0 support

The Bus Manager (stand-alone) now supports AUTOSAR files based on AUTOSAR 4.3.0 as communication matrices.

### New bus configuration features

The Bus Manager now provides additional bus configuration features:

- You can add the PDU Cyclic Timing Control feature to CAN TX PDUs. When you do this, you can specify a time period and/or time offset for the affected CAN TX PDUs independently from cyclic timings specified in the communication matrix.
- You can add the PDU RX Status feature to PDUs that are assigned to the **Inspection** part of bus configurations. When you do this, you can inspect the status of the affected received PDUs. For example, you can count the number of the PDU's receptions or provide the time of the receptions to a connected behavior model.

For more information, refer to [Working with Bus Configuration Features \(ConfigurationDesk Bus Manager Implementation Guide\)](#).

### New configurable communication matrix elements

The Bus Manager now lets you configure the following additional communication matrix elements:

- The length of CAN and LIN frames.
- The identifier, extended addressing, CAN FD support, and bit rate switch of CAN frames.
- The identifier and checksum type of LIN frames.
- The length and the unused bit pattern of PDUs.
- The endianness and start bit position of ISignal-to-IPDU mappings.


The specified settings apply to the affected element and all its instances in the active ConfigurationDesk application. For more information, refer to [Specifying User-Defined Settings for Communication Matrix Elements \(ConfigurationDesk Bus Manager Implementation Guide\)](#).



**New PDU elements to access supported PDU types**

The Bus Manager now provides additional PDU elements to access the supported PDU types. In addition to the **Bus ISignal IPDU**, **Bus Multiplexed IPDU**, and **Bus Container IPDU** elements, you can now also use the following elements:

- **Bus General-Purpose IPDU**
- **Bus General-Purpose PDU**
- **Bus DCM IPDU**
- **Bus NMPDU**
- **Bus NPDU**
- **Bus User-Defined IPDU**
- **Bus User-Defined PDU**


The PDU elements let you access related PDUs in browsers, tables, and via the automation interface, for example. For more information, refer to [Supported PDU Types and Signal Data Types](#) ( [ConfigurationDesk Bus Manager Implementation Guide](#)).

**Enhanced bus configuration tables**

The **Bus Configurations**, **Bus Simulation Features**, **Bus Inspection Features**, and **Bus Configuration Function Ports** tables now provide additional information on the communication matrix elements that are assigned to bus configurations. For example, the tables display all the clusters of which an assigned communication matrix element is a member and all the ECUs that transmit an assigned element. The displayed information is derived from the settings specified in the communication matrix and not determined by the actual element configuration in a bus configuration.

## Migrating to Bus Manager (Stand-Alone) 6.0

**Changes in the tool automation interface**

There are changes in the tool automation interface. Some of these changes affect the data model and can cause code from previous Releases to malfunction. For details, refer to [Changes to the Automation Interface for Release 2017-B](#) ( [ConfigurationDesk Automating Tool Handling](#)).

**Changes in TRC file for Bus Manager elements**

The paths of Bus Manager elements in the TRC file have changed from Bus Manager (stand-alone) 5.7 to Bus Manager (stand-alone) 6.0. When you generate bus simulation containers with Bus Manager (stand-alone) 6.0, you might have to adapt projects that use the generated TRC file (e.g., generate new instrument layouts in ControlDesk).



# ConfigurationDesk

---

**Two variants for different use scenarios**

ConfigurationDesk is provided in two variants that are useful for different use scenarios. You can use ConfigurationDesk - Implementation Version to implement real-time applications. You can use ConfigurationDesk - Configuration Version to configure dSPACE RapidPro hardware.

---

**Where to go from here****Information in this section**

<a href="#">ConfigurationDesk - Implementation Version .....</a>	<a href="#">84</a>
<a href="#">ConfigurationDesk - Configuration Version .....</a>	<a href="#">95</a>

# ConfigurationDesk - Implementation Version

## Where to go from here

## Information in this section

New Features of ConfigurationDesk 6.0 (Implementation Version) .....	84
Migrating to ConfigurationDesk 6.0 .....	92

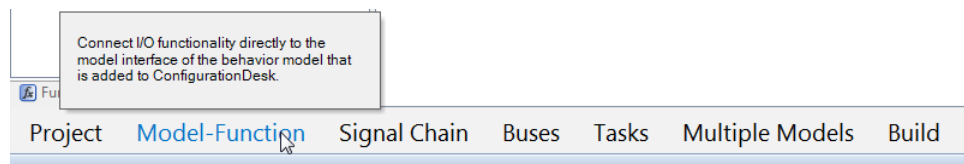
## New Features of ConfigurationDesk 6.0 (Implementation Version)

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### User interface consists of view sets for specific purposes

The user interface of ConfigurationDesk now offers different view sets for specific purposes. You can switch between view sets by using the navigation bar:

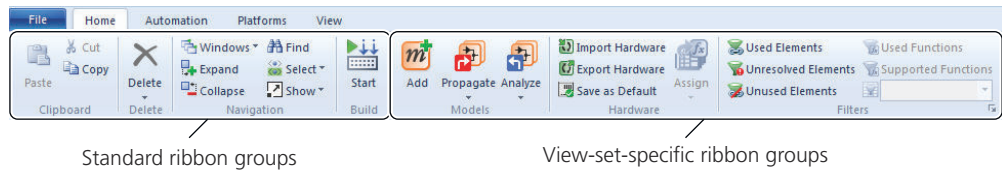


The available panes of each view set serve a specific purpose. For example, the **Project** view set contains the **Project Manager** to perform project and application management tasks.

The order of view sets from left to right on the navigation bar represents the workflow for implementing a real-time application. The **Project** and **Build** view sets are the start and end points for all use scenarios. The other view sets are suitable for specific use scenarios.

For each view set, the **Home** ribbon contains specific commands suitable for the purpose of the view set.

The following illustration shows the view set-specific **Home** ribbon of the **Model-Function** view set:



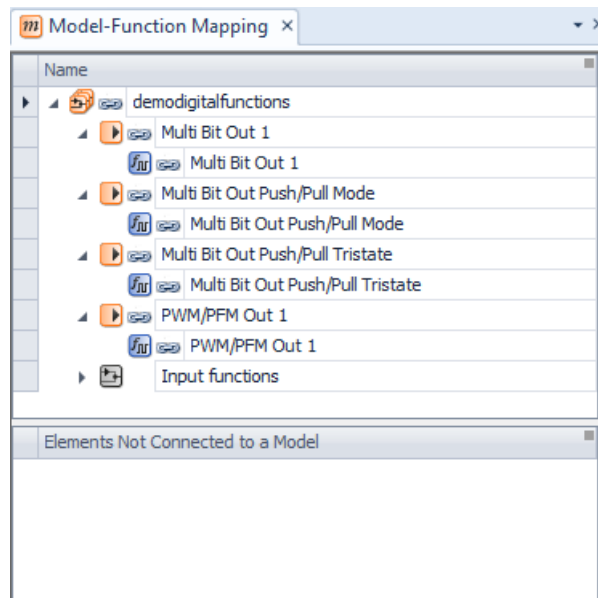
If the available view sets do not meet your requirements, you can customize them or create additional view sets with the panes of your choice.

For details and instructions regarding the handling and the customization of ConfigurationDesk's user interface, refer to [User Interface of ConfigurationDesk](#) ([ConfigurationDesk Real-Time Implementation Guide](#)).

## New Model-Function Mapping Browser

ConfigurationDesk now provides the **Model-Function Mapping Browser** for a simplified connection of ConfigurationDesk and Simulink behavior models. The **Model-Function Mapping Browser** has the following benefits:

- It shows the structure of the behavior models and the I/O functionality at a glance.
- It lets you quickly create and update signal chains for the work with Simulink behavior models.
- It provides new commands, that let you update the ConfigurationDesk model interface and the Simulink model interface based on the configuration of the I/O functionality.



**Simplified creation of signal chains** The **Model-Function Mapping Browser** displays all the model implementations including their subsystems of the active ConfigurationDesk application. You can easily create signal chains by dragging and dropping hardware channels or function blocks to a Simulink

behavior model or its subsystem. If you do this, ConfigurationDesk automatically creates model port blocks with suitable configurations.

**Simplified modeling of asynchronous tasks** In addition to the model port blocks for data inports and data outports, ConfigurationDesk now creates **Runnable Function** blocks for function blocks that provide an I/O event. **Runnable Function** blocks provide runnable functions that are used for modeling asynchronous tasks. If you map a Runnable Function block to a function block with enabled event generation, ConfigurationDesk creates a task that has a runnable function and the I/O event assigned.

**Elements not connected to a model** To complete the overview, the **Model-Function Mapping Browser** provides the **Elements Not Connected to a Model** list that displays all model port blocks and function blocks that are not connected to a model.

**Propagating changes to a Simulink model** Modifications that you made in the **Model-Function Mapping Browser** can be propagated directly to a Simulink behavior model.

**Note**

Propagating changes to a Simulink model can be applied to all signal chains. You can select which signal chains must be propagated to the Simulink model.

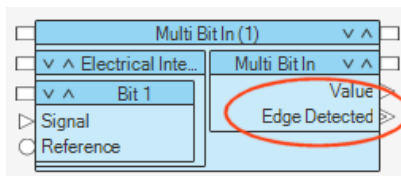
For more information, refer to:

- [Simplified Connection of ConfigurationDesk and Simulink Models](#) (📖 ConfigurationDesk Real-Time Implementation Guide).
- [📖 ConfigurationDesk Tutorial Starting with Simulink.](#)


**Runnable function blocks and event ports for modeling asynchronous tasks**

ConfigurationDesk now provides the following new features that let you easily model asynchronous tasks:

- For function blocks with enabled event generation, an event port is now shown at the function block.



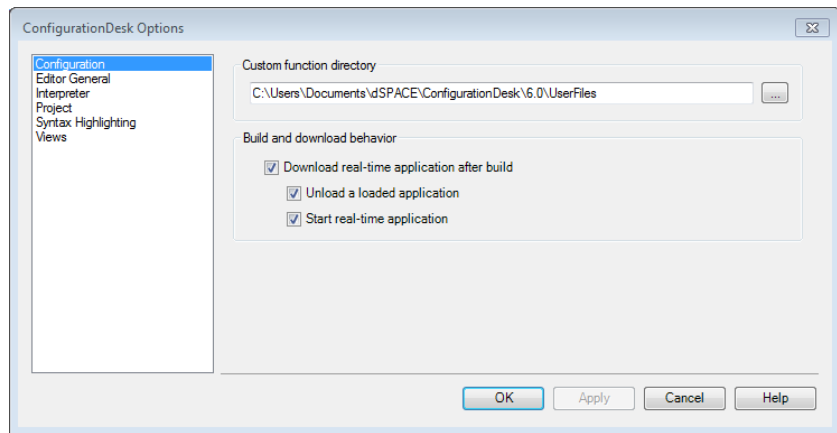
- ConfigurationDesk now provides **Runnable Function** blocks as model port blocks. The **Runnable Function** blocks are displayed in ConfigurationDesk as shown in the following table:

Browser	Graphical Display
<b>Model-Function Mapping Browser and Model Browser</b>	
<b>Signal Chain Browser</b>	<div style="border: 1px solid black; padding: 2px;"> <input type="checkbox"/> v ^ Multi Bit In (1)\Edge Detected                      demodigitalfunctions ProcessingUnitAp...  <input checked="" type="checkbox"/> &gt; Multi Bit In (1)\Edge Detected                 </div>

You can now create and model asynchronous tasks easily by mapping the event port to the port of the **Runnable Function** block. The task priorities can be specified directly in the properties of the **Runnable Function** block without having to change to the **Task Configuration** table. For details, refer to [Modeling Asynchronous Tasks](#) ( [ConfigurationDesk Real-Time Implementation Guide](#)).

**New option for default download behavior**

The **ConfigurationDesk Options** dialog provides a new option to configure the default download behavior.



**Predefined hardware topologies available**

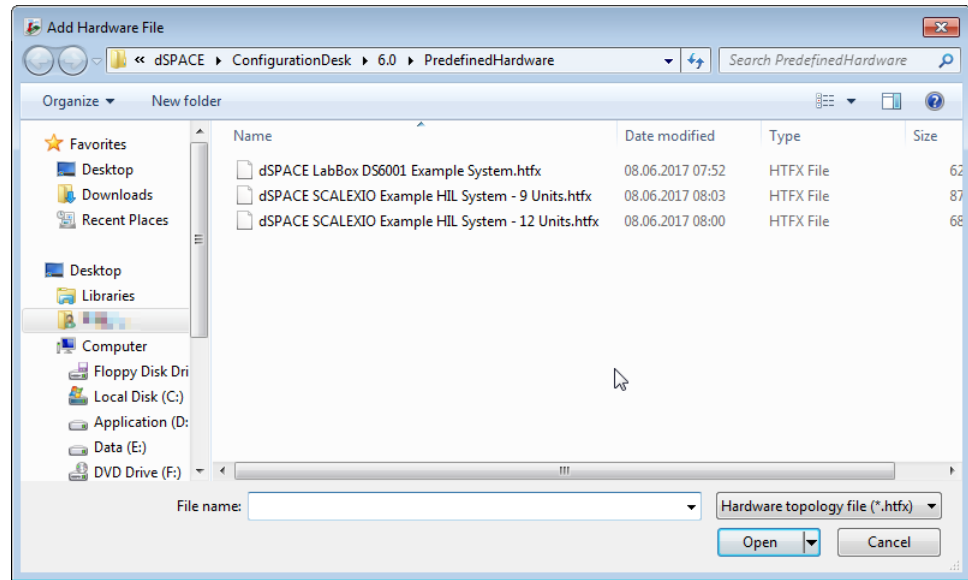
ConfigurationDesk now offers a number of predefined hardware topologies that you can add to your ConfigurationDesk application.

The topologies are stored in HTFX files in the following folder:

My Documents\dSPACE\ConfigurationDesk\

<x.y> is a placeholder for the software version.

The folder opens by default when you open the **Add Hardware File** dialog, for example, when you are adding hardware to a new ConfigurationDesk application.



**Supported SIC file versions**

ConfigurationDesk 6.0 supports SIC file versions as listed below:

SIC Files Created with Model Interface Package for Simulink of ...	SIC Version
dSPACE Release 2017-B (Model Interface Package for Simulink 3.5)	1.3
dSPACE Release 2017-A (Model Interface Package for Simulink 3.4)	1.2.1
dSPACE Release 2016-B (Model Interface Package for Simulink 3.3)	1.2
dSPACE Release 2016-A (Model Interface Package for Simulink 3.2)	1.1

**Supported V-ECU implementation container versions**

ConfigurationDesk 6.0 supports V-ECU implementation container versions as listed below.:

V-ECU Implementations Created With...	V-ECU Implementation Version
dSPACE Release 2017-B: <ul style="list-style-type: none"> <li>▪ SystemDesk 5.0</li> <li>▪ TargetLink 4.3</li> </ul>	2.6
dSPACE Release 2017-A: <ul style="list-style-type: none"> <li>▪ SystemDesk 4.8</li> </ul>	2.5
dSPACE Release 2016-B: <ul style="list-style-type: none"> <li>▪ SystemDesk 4.7</li> <li>▪ TargetLink 4.2</li> </ul>	2.4.1
dSPACE Release 2016-A: <ul style="list-style-type: none"> <li>▪ SystemDesk 4.6</li> </ul>	2.4



---

### New features of the FMU support

- ConfigurationDesk now supports the assignment of multiple FMUs to the same application process. This support applies only for FMUs, not to any other model implementation type (for example, Simulink model implementations). For more information, refer to [Using Multiple FMUs in the Same Application Process](#) ([ConfigurationDesk Real-Time Implementation Guide](#)).
- ConfigurationDesk supports FMUs with source files that use functions and/or variables of compiled binaries. As of dSPACE Release 2017-B, ConfigurationDesk includes these binaries automatically during the build process if they reside in a specific folder. For more information, refer to [Preconditions for Using FMUs in ConfigurationDesk](#) ([ConfigurationDesk Real-Time Implementation Guide](#)).  
ConfigurationDesk also supports the precompilation of FMUs with source files and included binaries.

---

### New features of the bus simulation container support

ConfigurationDesk now lets you map several **Configuration** ports of a bus simulation container to one **CAN** or **LIN** function block. When you do this, you can assign the **Configuration** ports to the same hardware resource. This is useful, for example, if the **Configuration** ports were generated for bus access requests that represent the same communication cluster.

For more information, refer to [Handling Bus Simulation Containers in a ConfigurationDesk Application](#) ([ConfigurationDesk Real-Time Implementation Guide](#)).

---

### New function block types

**Digital Incremental Encoder In** The **Digital Incremental Encoder In** function block measures the speed and position based on the signals of a connected rotary or linear incremental encoder.

For more information, refer to [Digital Incremental Encoder In](#) ([ConfigurationDesk I/O Function Implementation Guide](#))

**Multi-Channel PWM Out** The **Multi-Channel PWM Out** function block type synchronously generates multiple PWM signals with a common frequency. The function block can work as a provider: Other function blocks can use the generated trigger signal as a trigger source.

For more information, refer to [Multi-Channel PWM Out](#) ([ConfigurationDesk I/O Function Implementation Guide](#))

**Digital Pulse Out** The **Digital Pulse Out** function block generates a digital pulse with each sampling step of the behavior model tasks or with each trigger event of another function block.

For more information, refer to [Digital Pulse Out](#) ([ConfigurationDesk I/O Function Implementation Guide](#))

---

**Enhanced function block type**    **Wheelspeed Out**    The **Wheelspeed Out** function block type now supports the following new features:

- Simulates predefined gaps in the coding of the wheel. Wheel speed sensors with predefined gaps in the coding of the sensor wheel are used to define an index position. For example, crankshaft sensors commonly use this method.
- Provides an I/O event each time a new revolution starts or a pitch is detected.
- Lets you configure the electrical characteristics of the generated pulses from within the behavior model via function ports.

For more information, refer to [Wheelspeed Out](#) ([📖 ConfigurationDesk I/O Function Implementation Guide](#)).

---

**New features of the ECU interfacing support**

ConfigurationDesk now lets you implement the handling of ECU calibration pages and the data access for individual ECU variables in a real-time application. If prepared with the ECU Interface Manager, you can perform the following actions, for example:

- Switch the active ECU calibration page at run-time.
- Enable reading or writing individual ECU variables from or to the ECU application.

For more information, refer to [Introduction to ECU Interfacing with SCALEXIO Systems](#) ([📖 ConfigurationDesk Real-Time Implementation Guide](#)).

---

**New features of the Bus Manager**

**AUTOSAR 4.3.0 support**    The Bus Manager now supports AUTOSAR files based on AUTOSAR 4.3.0 as communication matrices.

**New bus configuration features**    The Bus Manager now provides additional bus configuration features:


- You can add the PDU Cyclic Timing Control feature to CAN TX PDUs. When you do this, you can specify a time period and/or time offset for the affected CAN TX PDUs independently from cyclic timings specified in the communication matrix.
- You can add the PDU RX Status feature to PDUs that are assigned to the **Inspection** part of bus configurations. When you do this, you can inspect the status of the affected received PDUs. For example, you can count the number of the PDU's receptions or provide the time of the receptions to a connected behavior model.

For more information, refer to [Working with Bus Configuration Features](#) ([📖 ConfigurationDesk Bus Manager Implementation Guide](#)).

**New configurable communication matrix elements**    The Bus Manager now lets you configure the following additional communication matrix elements:


- The length of CAN and LIN frames.
- The identifier, extended addressing, CAN FD support, and bit rate switch of CAN frames.
- The identifier and checksum type of LIN frames.

- The length and the unused bit pattern of PDUs.
- The endianness and start bit position of ISignal-to-IPDU mappings.

The specified settings apply to the affected element and all its instances in the active ConfigurationDesk application. For more information, refer to [Specifying User-Defined Settings for Communication Matrix Elements](#) ( [ConfigurationDesk Bus Manager Implementation Guide](#)).

**New PDU elements to access supported PDU types** The Bus Manager now provides additional PDU elements to access the supported PDU types. In addition to the **Bus ISignal IPDU**, **Bus Multiplexed IPDU**, and **Bus Container IPDU** elements, you can now also use the following elements:

- **Bus General-Purpose IPDU**
- **Bus General-Purpose PDU**
- **Bus DCM IPDU**
- **Bus NMPDU**
- **Bus NPDU**
- **Bus User-Defined IPDU**
- **Bus User-Defined PDU**


The PDU elements let you access related PDUs in browsers, tables, and via the automation interface, for example. For more information, refer to [Supported PDU Types and Signal Data Types](#) ( [ConfigurationDesk Bus Manager Implementation Guide](#)).

**Enhanced bus configuration tables** The **Bus Configurations**, **Bus Simulation Features**, **Bus Inspection Features**, and **Bus Configuration Function Ports** tables now provide additional information on the communication matrix elements that are assigned to bus configurations. For example, the tables display all the clusters of which an assigned communication matrix element is a member and all the ECUs that transmit an assigned element. The displayed information is derived from the settings specified in the communication matrix and not determined by the actual element configuration in a bus configuration.

---

### New FPGA UART demo project

The demo project **CfgFPGAuartDemo** lets you implement a configurable UART bus communication on a DS2655 FPGA Base Board with a DS2655M2 Digital I/O Module. Knowledge on FPGA programming is not necessary to use the example or to reuse the example in your ConfigurationDesk project.

For more information, refer to [Building the Signal Chain for UART Communication Using an FPGA Board](#) ( [ConfigurationDesk Real-Time Implementation Guide](#)).

---

### New features concerning hardware support


ConfigurationDesk supports the following new SCALEXIO hardware:

- DS6202 Digital I/O Board

The DS6202 Digital I/O Board provides 32 fast bidirectional channels for advanced I/O functions. These channels can also be configured in pairs to establish up to 16 differential inputs.

- DS6311 FlexRay Board  
The DS6311 FlexRay Board provides four FlexRay controllers with two FlexRay channels (A and B) each.
- DS6341 CAN Board  
The DS6341 LIN Board provides 4 independent CAN/CAN FD channels.
- DS6351 LIN Board  
The DS6351 LIN Board provides 8 independent LIN channels.
- DS6551 IOCNET Link Board  
The DS6551 IOCNET Link Board provides an additional optical IOCNET port in a SCALEXIO LabBox. It converts one electrical IOCNET port of the box's backplane to an optical IOCNET port.

#### New features of the tool automation interface

ConfigurationDesk's automation interface supports additional features of ConfigurationDesk. For more information, refer to [Changes to the Automation Interface for Release 2017-B](#) ( [ConfigurationDesk Automating Tool Handling](#)).

## Migrating to ConfigurationDesk 6.0

#### Runnable functions from projects created with ConfigurationDesk 5.7 or earlier

If you work with ConfigurationDesk projects created with ConfigurationDesk 5.7 or earlier, the following applies:

If the ConfigurationDesk project contains runnable functions from the Simulink model, the related **Runnable Function** blocks are displayed on the model root level in the **Model-Function Mapping Browser**, even if the **Runnable Function** blocks are not located on the root level of the Simulink behavior model. The name you specified for the runnable function in the **Runnable Function** block dialog in Simulink is displayed as the **Runnable Function** block name in the **Model-Function Mapping Browser**. You have to perform a model analysis of the Simulink model. The **Model-Function Mapping Browser** then displays these **Runnable Function** blocks with the correct name and with the correct location in the model hierarchy.

#### Note

This also applies to Simulink implementation container files (SIC files) that have been created with the Model Interface Package for Simulink 3.4 or earlier.

#### Propagating changes to Simulink

Propagating changes to a Simulink model overwrites specific configurations in the Simulink model, such as the port configuration of **Data Inport** blocks and **Data Outport** blocks, and the settings on the **Runnable Function** page of the **Runnable Function** block dialog. These parameters are configured based on the configuration of the function blocks to which they are mapped.

If you work with ConfigurationDesk projects created with ConfigurationDesk 5.7 or earlier, keep in mind that propagating changes to Simulink affects all the signal chains, including signal chains that have been created manually with an older ConfigurationDesk version.

#### Tip

To prevent parts of a Simulink model from being accidentally changed by a propagate operation, you can protect specific subsystems by setting them to read-only in Simulink. When analyzed, these subsystems are marked with a lock symbol in ConfigurationDesk.

### ConfigurationDesk projects containing FlexRay communication

As of ConfigurationDesk 6.0, function blocks have event ports for I/O events. If you work with ConfigurationDesk projects that contain FlexRay communication and that were created with ConfigurationDesk 5.7 or earlier, the following applies: Each instantiated **FlexRay** function block provides event ports, regardless of whether you use the related I/O events in your Simulink model:

- Event ports related to I/O events that you use in your Simulink model are automatically mapped to Runnable Function blocks. In this case, you must analyze your Simulink model. For more information, refer to [Runnable functions from projects created with ConfigurationDesk 5.7 or earlier](#) on page 92.
- Event ports related to I/O events that you do not use in your Simulink model are unmapped. You cannot disable or delete event ports from **FlexRay** function blocks. Instead, you can leave the event ports unmapped. Keep in mind that each time you propagate changes to Simulink, Runnable Function blocks are automatically created for these event ports. For more information, refer to [Propagating changes to Simulink](#) on page 92.

### Changes to the tool automation interface

Some changes to the tool automation interface affect the data model and can cause code from previous Releases to malfunction. For more information, refer to [Changes to the Automation Interface for Release 2017-B](#) ( [ConfigurationDesk Automating Tool Handling](#)).

### Inconsistency with Ethernet adapters

When you migrate a project of dSPACE Release 2016-A or earlier, the migration process adds the Ethernet adapter of the SCALEXIO Real-Time PC to the hardware topologies of the migrated project. The default name of the added Ethernet adapter might not match the Ethernet adapter name of the accessible platforms. Therefore, the status bar shows the status "No matching platform connected". This status prevents the automatic download of the real-time application after the build. However, you can build the real-time application and manually download it to the hardware.

To resolve the inconsistency, specify an identical name for the Ethernet adapter of the SCALEXIO Real-Time PC in the **Hardware Resource Browser** and in the **Platform Manager**, e.g., by replacing the hardware topology.

**Changes in TRC file for Bus Manager elements**

The paths of Bus Manager elements in the TRC file have changed from ConfigurationDesk 5.7 to ConfigurationDesk 6.0. When you build a real-time application or generate bus simulation containers with ConfigurationDesk 6.0, you might have to adapt projects that use the generated TRC file (e.g., generate new instrument layouts in ControlDesk).

# ConfigurationDesk - Configuration Version

## Introduction

With ConfigurationDesk's Configuration version, you can configure the RapidPro hardware.

## New Features of ConfigurationDesk 6.0 (Configuration Version)

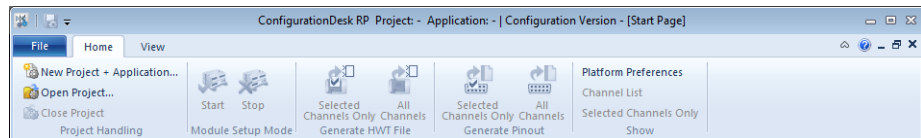
### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### More intuitive user interface

The user interface of ConfigurationDesk - Configuration Version is more intuitive: Its menu bar and toolbars have been replaced by ribbons and the Backstage view as used in ControlDesk, Microsoft Office, etc.

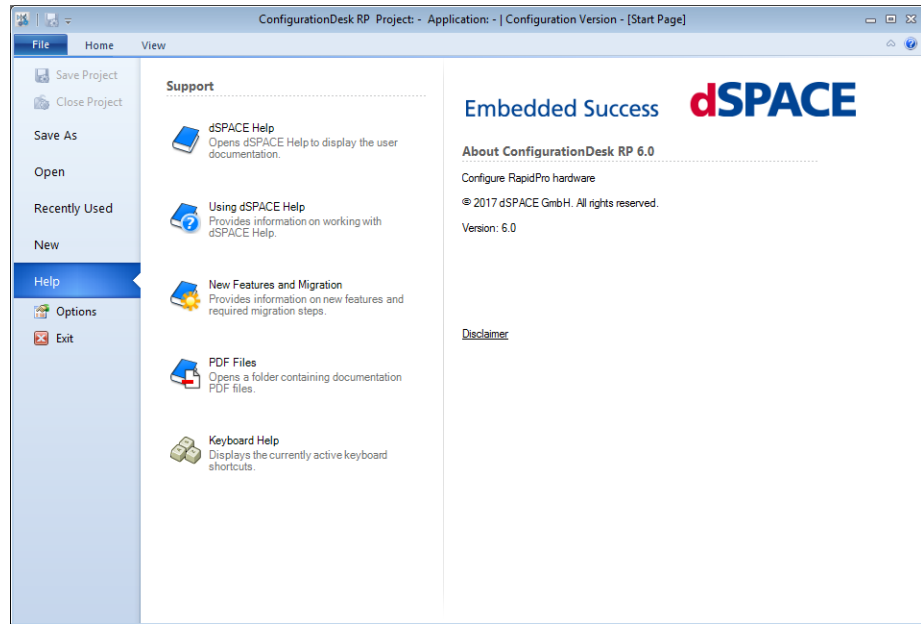
**Ribbon** ConfigurationDesk's ribbons organize and group commands that belong together. They are located at the top of the user interface. Refer to the following illustration:



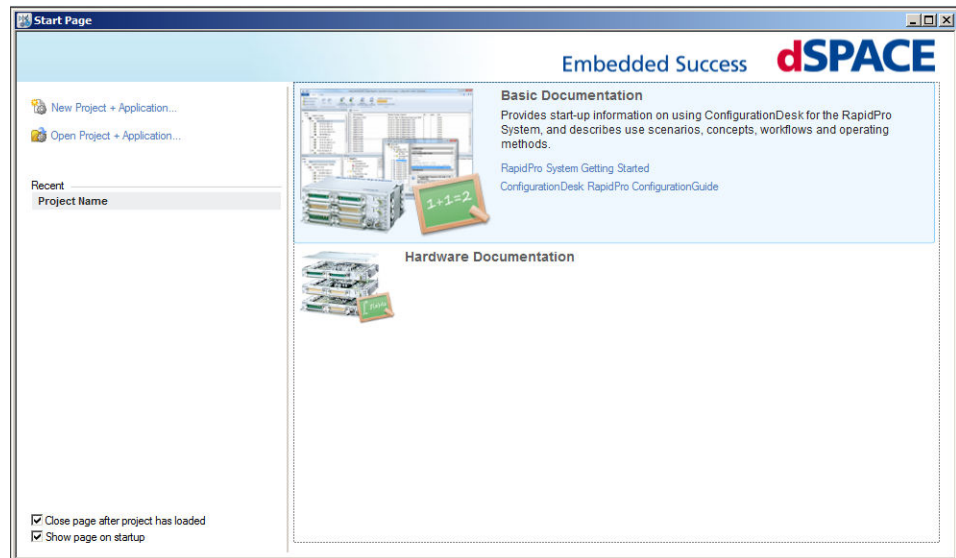
Each ribbon has ribbon groups, each of which provides a set of related commands. For example, the Home ribbon contains the **Project Handling**, **Module Setup Mode**, **Generate HWT File** ribbon groups.

**Backstage view** ConfigurationDesk's Backstage view provides basic commands, for example, for opening, importing, and saving projects. It also provides quick access to the recently used projects and experiments.

The following illustration shows the Backstage view with the **Help** ribbon group:



**Start page** ConfigurationDesk's Start page provides quick access to the most recently opened projects and to the user documentation of ConfigurationDesk - Configuration Version.





# ControlDesk

---

**Where to go from here**

**Information in this section**

New Features of ControlDesk 6.2 .....	98
Migrating to ControlDesk 6.2 .....	113

## New Features of ControlDesk 6.2

### Where to go from here

### Information in this section

<a href="#">New Features of Platform Management and Platforms/Devices (ControlDesk 6.2)</a> .....	98
Gives an overview of new features of platform management and platforms/devices for ControlDesk 6.2.	
<a href="#">New Variable Management Features (ControlDesk 6.2)</a> .....	100
Gives an overview of the new variable management features of ControlDesk 6.2.	
<a href="#">New Instrument Features (ControlDesk 6.2)</a> .....	102
Gives an overview of the new instrument features as of ControlDesk 6.2.	
<a href="#">New Measurement and Recording Features (ControlDesk 6.2)</a> .....	105
Gives an overview of the new measurement and recording features as of ControlDesk 6.2.	
<a href="#">New Automation Features (ControlDesk 6.2)</a> .....	105
Gives an overview of the new automation interface features as of ControlDesk 6.2.	
<a href="#">New Bus Navigator Features (ControlDesk 6.2)</a> .....	106
Gives an overview of the new Bus Navigator features as of ControlDesk 6.2.	
<a href="#">New Signal Editor Features (ControlDesk 6.2)</a> .....	109
Gives an overview of the new Signal Editor features as of ControlDesk 6.2.	
<a href="#">Further Enhancements and Changes with ControlDesk (ControlDesk 6.2)</a> .....	110
Gives an overview of further enhancements as of ControlDesk 6.2.	

## New Features of Platform Management and Platforms/Devices (ControlDesk 6.2)

### New Ethernet Bus Monitoring device

ControlDesk now provides the *Ethernet Bus Monitoring device*. You can use the device to monitor and log Ethernet communication in connection with ControlDesk's Bus Navigator.

**Note**

Keep in mind the important information when monitoring and logging Ethernet traffic with ControlDesk.

Refer to [Important Information when Monitoring and Logging Ethernet Traffic](#) ([📖 ControlDesk Introduction and Overview](#)).

For instructions on configuring the device, refer to [How to Configure an Ethernet Bus Monitoring Device](#) ([📖 ControlDesk Platform Management](#)).

---

**XIL API MAPort platform:  
Example showing the access  
to third-party hardware**

ControlDesk provides an example of a custom XIL API MAPort implementation. You can adapt the example easily to access third-party hardware via ControlDesk.

Refer to [Example of Using a Custom XIL API MAPort Implementation](#) ([📖 ControlDesk Platform Management](#)).

---

**ECU Diagnostics device:  
Support of CAN channels of  
dSPACE SCALEXIO and VEOS**

The ECU Diagnostics device now supports CAN channels of dSPACE platforms.

The following CAN channels are supported as CAN interfaces:

- CAN channels of a dSPACE SCALEXIO system
- CAN channels on VEOS

Refer to [Supported CAN Interfaces](#) ([📖 ControlDesk Platform Management](#)).

---

**Reloading a real-time  
application to the flash  
memory**

ControlDesk now lets you reload a real-time application to the flash memory.

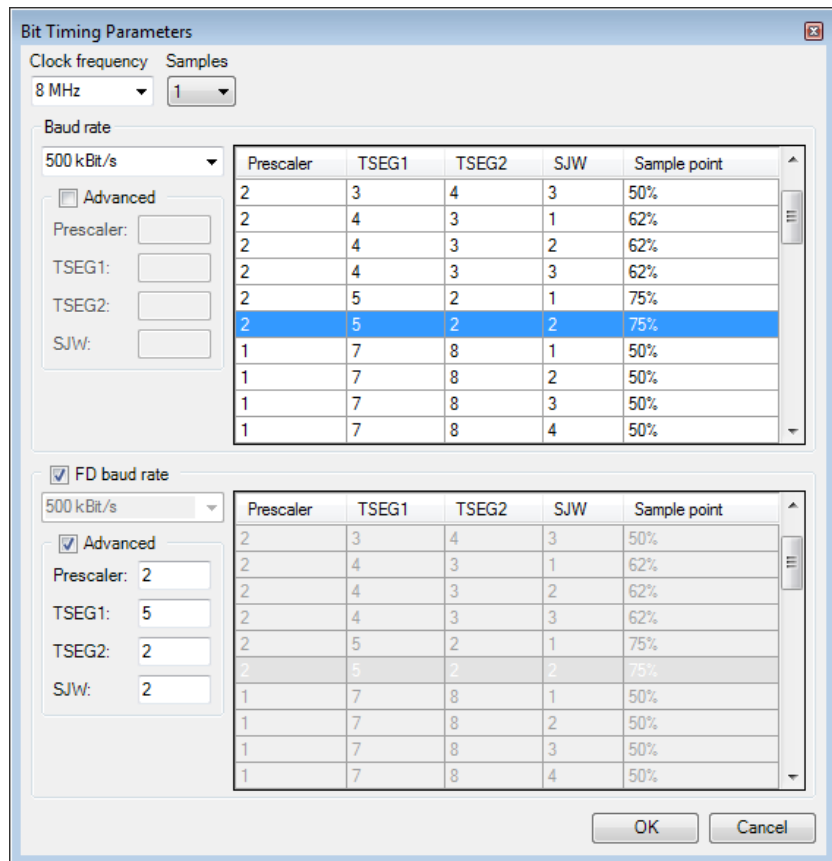
For details, refer to:

- DS1007, DS1202 MicroLabBox, and SCALEXIO:
  - [Reload to Flash](#) ([📖 ControlDesk Platform Management](#))
  - [Reload to Flash and Start](#) ([📖 ControlDesk Platform Management](#))
- DS1005, DS1006, DS1104, and MicroAutoBox:
  - [Real-Time Application - Reload to Flash](#) ([📖 ControlDesk Platform Management](#))

---

**Improved configuration of  
CAN bit timing parameters**

ControlDesk now provides the **Bit Timing Parameters** dialog for you to easily configure CAN bit timing parameters of CAN-based devices. You can specify the parameters for CAN and CAN FD separately as shown in the following illustration.



Refer to [CAN Settings Properties](#) ([ControlDesk Platform Management](#)).

## Related topics

### Basics

[Important Information when Monitoring and Logging Ethernet Traffic](#)  
([ControlDesk Introduction and Overview](#))

## New Variable Management Features (ControlDesk 6.2)

### A2L file import: Support for variable-specific IF\_DATA

ControlDesk now lets you specify whether to use variable-specific interface description data (IF\_DATA). If the use of variable-specific interface description data is enabled, measuring and recording with ControlDesk support variable-specific default rasters as defined in the A2L file's IF\_DATA.

Refer to [Variables Page](#) ([ControlDesk Variable Management](#)).

---

**A2L file import: Ignoring variables at the address 0x0**

ControlDesk now lets you specify whether to ignore variables at the address **0x0** during the import of an A2L file.

In early ECU development stages, ECU variables that are not implemented yet can have the address **0x0** in the A2L file of the related ECU application. If you ignore these ECU variables during the A2L file import, they cannot be mistakenly calibrated.


Refer to [Variables Page](#) ( [ControlDesk Variable Management](#)).

---

**AUTOSAR/FIBEX file import: Multicenter files supported by bus monitoring devices**

AUTOSAR and FIBEX files can contain the description of *multiple clusters*. A cluster is a communication network of network nodes that are connected to the same physical channels and share the same bus protocol and address range.

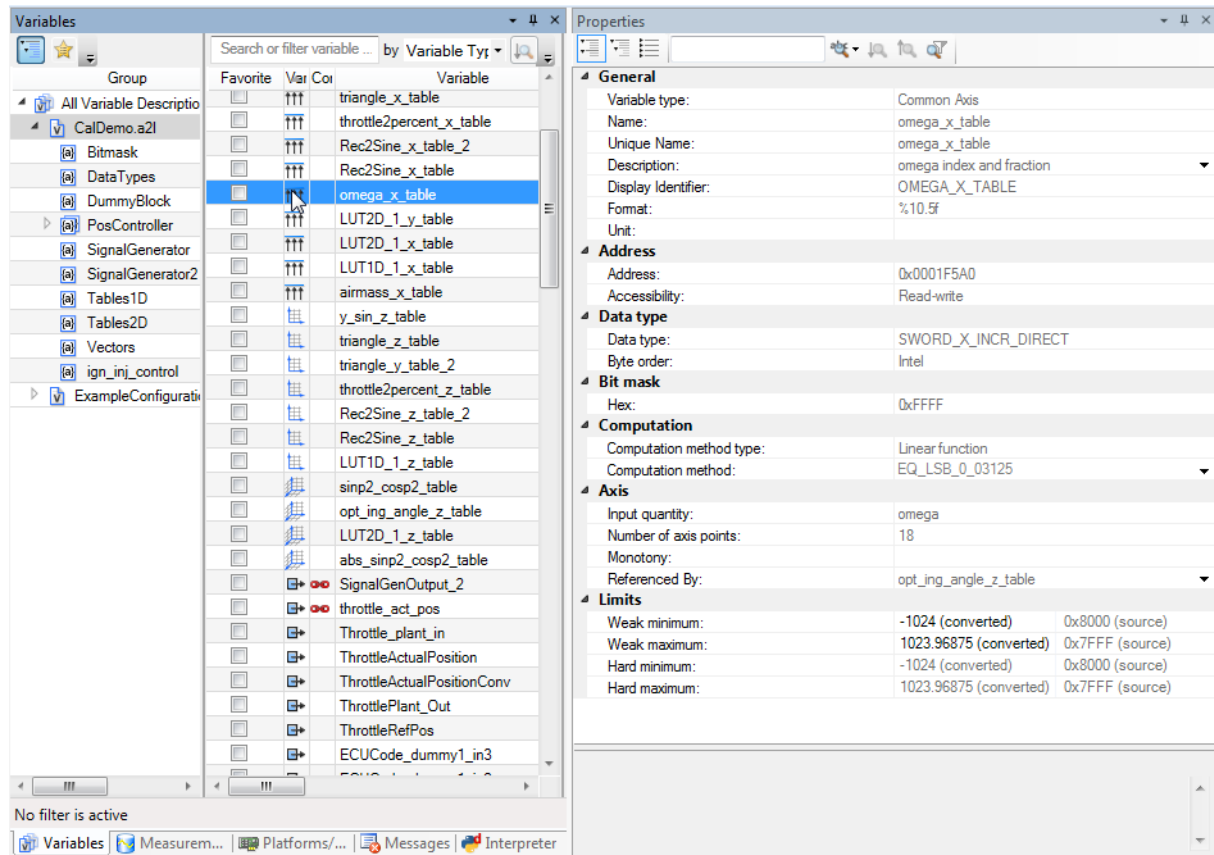
When you add a variable description file to a bus monitoring device, ControlDesk 6.2 now lets you select the cluster to be imported.

For instructions, refer to [How to Add a Variable Description to a Platform/Device](#) ( [ControlDesk Variable Management](#)).

---

**Display of variable properties in the Properties controlbar**

ControlDesk now displays the properties of variables in the **Properties** controlbar. As an example, the following illustration shows the properties of the **omega\_x\_table** variable selected in the **Variable Browser**.



Refer to [Properties of Variables \(Variable Browser\)](#) (ControlDesk Variable Management).

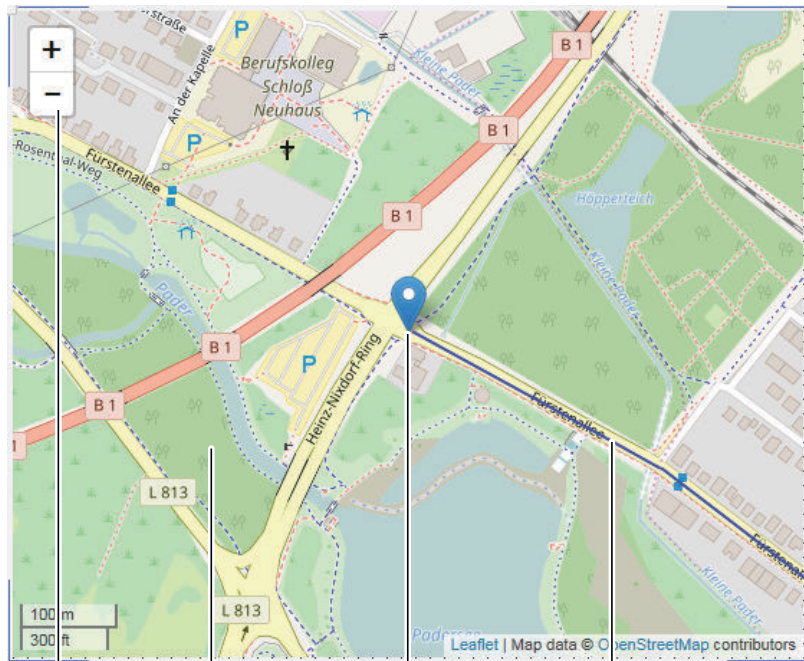
## Support of Int64 and UInt64 variables in TRC files

ControlDesk now supports Int64 and UInt64 variables in TRC files.

## New Instrument Features (ControlDesk 6.2)

### New Map instrument

ControlDesk now provides the Map instrument. You can use the Map instrument to display GPS motion data. For example, this allows you to display the motion of a car in relation to the time stamps of recorded signals.



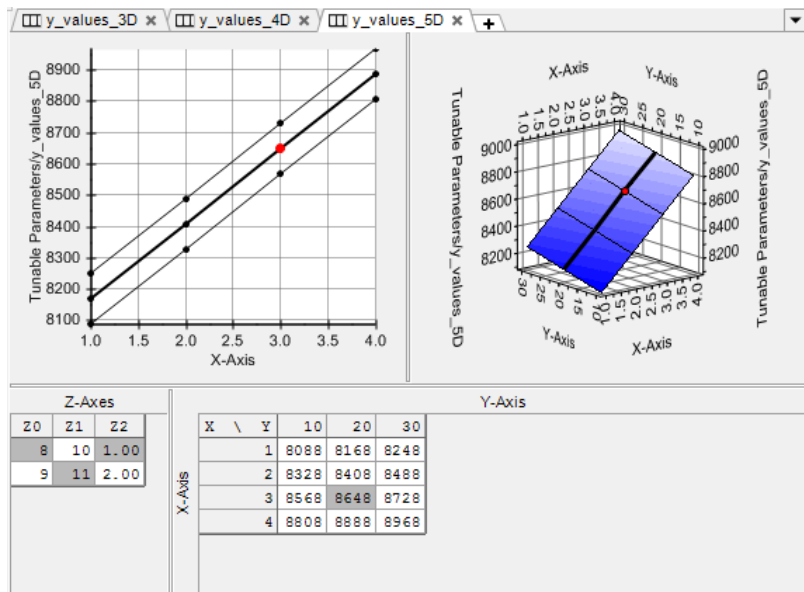
Zoom Content area Marker Route

Refer to [Map](#) (ControlDesk Instrument Handling).

**Table Editor: Connecting multidimensional table data (n-D tables)**

The Table Editor can now display multidimensional table data (n-D tables). The x- and y-axis represent the first and second dimension. For each additional dimension, a z-axis is added to a separate z-axis area in the grid view.

The following illustration shows an example with 5-D table data:



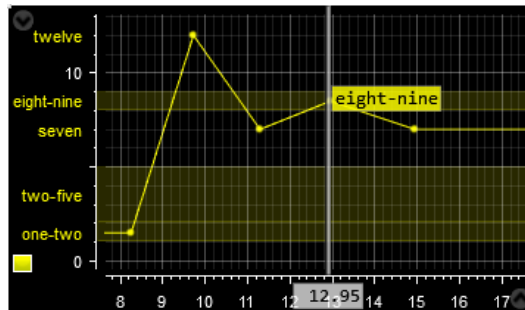
For instructions on how to connect multidimensional table data in a Table Editor, refer to [How to Connect Multidimensional Table Data to the Table Editor](#) ([ControlDesk Instrument Handling](#)).

### Time Plotter/Index Plotter: Support of variables using conversion tables

The Time Plotter and the Index Plotter now support variables using conversion tables. Converted values can now be displayed on the y-axis and on the time cursor. Multiscalings are also supported.

Ranges of a verbal range conversion table can be visualized in the chart.

The following illustration shows an example:



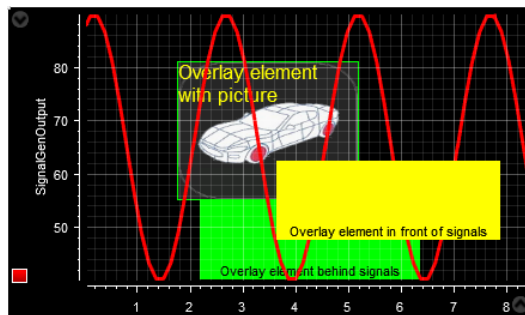
Refer to:

- [Basics of Handling the Time Plotter](#) ([ControlDesk Instrument Handling](#))
- [Basics of Handling the Index Plotter](#) ([ControlDesk Instrument Handling](#))

### Time Plotter/Index Plotter: Displaying overlay elements

The Time Plotter and the Index Plotter now let you place overlay elements with text or pictures in front of or behind the signals.

The following illustration shows an example:



You can fix an overlay element in the following ways:

- You can fix the element to a chart position so it moves with the chart.
- You can fix the element to an instrument position so it does not move with the chart.

Refer to:

- [Basics of Handling the Time Plotter](#) ([ControlDesk Instrument Handling](#))
- [Basics of Handling the Index Plotter](#) ([ControlDesk Instrument Handling](#))



## New Measurement and Recording Features (ControlDesk 6.2)

### Support of variable-specific default rasters

ControlDesk now lets you specify whether to use variable-specific interface description data (IF\_DATA) including variable-specific raster information; see [New Variable Management Features \(ControlDesk 6.2\)](#) on page 100.

If the use of variable-specific interface description data is enabled, measurement and recording with ControlDesk support variable-specific default rasters as defined in the A2L file's IF\_DATA.

Refer to [Variables Page \(ControlDesk Variable Management\)](#).

### Incremental saving of MF4 files

ControlDesk 6.2 now supports *incremental saving* of MF4 files: Modifications to an MF4 file are saved much faster as long as you do not remove signals from the MF4 file.

Incremental saving is supported when you modify the following file contents:

- Description
- X-axis offset
- MDF properties (department, project, measurement object)
- Bookmarks

Information on the signals, data, and capture blocks remain unchanged during incremental saving.

## New Automation Features (ControlDesk 6.2)

### Adding parameters to a sub data set

As of version 6.1, ControlDesk lets you add parameters to a sub data set via the user interface.

ControlDesk 6.2 now lets you add parameters to a sub data set also via its automation interface.

Refer to [DataSetParameters / IXaDataSetParameters <<Collection>>](#) ([ControlDesk Automation](#)).

### Removing parameters from a sub data set

As of version 6.1, ControlDesk lets you remove parameters from a sub data set via the user interface.

ControlDesk 6.2 now lets you remove parameters from a sub data set also via its automation interface.

Refer to [DataSetParameters / IXaDataSetParameters <<Collection>>](#) ([ControlDesk Automation](#)).

## New Bus Navigator Features (ControlDesk 6.2)

### Ethernet monitoring and logging

ControlDesk now provides the *Ethernet Bus Monitoring device*. You can use the device to monitor and log Ethernet communication in connection with ControlDesk's Bus Navigator.

The following illustration shows an example.

Identifier	Monitor Time	Version	IHL	Differentiate	Explicit Con.	Total Length	Identification	Flags	Fragment Of.	Time To Live	Protocol	Header Che.	Sc
Loopb...	1277.4709600	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	10481 (0x...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1279.0742020	4	24 bytes...	CS0 (0x00)	Not-ECT (...)	32	360 (0x16...	0x00 (Res... 0	0	1	IGMP (2)	0x0000	10
Loopb...	1279.5729700	4	24 bytes...	CS0 (0x00)	Not-ECT (...)	32	361 (0x16...	0x00 (Res... 0	0	1	IGMP (2)	0x0000	10
Loopb...	1280.4713920	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	365 (0x16...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1280.4716230	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	10482 (0x...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1282.5244780	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	56	10483 (0x...	0x04 (Res... 0	0	128	TCP (6)	0x0000	10
Loopb...	1282.5245110	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	40	10484 (0x...	0x04 (Res... 0	0	128	TCP (6)	0x0000	10
Loopb...	1282.5245260	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	249	10485 (0x...	0x04 (Res... 0	0	128	TCP (6)	0x0000	10
Loopb...	1282.5245450	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	40	10486 (0x...	0x04 (Res... 0	0	128	TCP (6)	0x0000	10
Loopb...	1282.5268960	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	264	10487 (0x...	0x04 (Res... 0	0	128	TCP (6)	0x0000	10
Loopb...	1282.5269100	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	40	10488 (0x...	0x04 (Res... 0	0	128	TCP (6)	0x0000	10
Loopb...	1283.0730800	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	172	397 (0x18...	0x00 (Res... 0	0	128	ICMP (1)	0x0000	10
Loopb...	1283.0731610	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	172	398 (0x18...	0x00 (Res... 0	0	128	ICMP (1)	0x0000	10
Loopb...	1283.4710840	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	400 (0x19...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1283.4711710	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	10489 (0x...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1286.5317550	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	404 (0x19...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1286.5320030	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	10490 (0x...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1289.0732570	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	172	405 (0x19...	0x00 (Res... 0	0	128	ICMP (1)	0x0000	10
Loopb...	1289.0733450	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	172	406 (0x19...	0x00 (Res... 0	0	128	ICMP (1)	0x0000	10
Loopb...	1289.5321110	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	407 (0x19...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1289.5322090	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	10491 (0x...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1290.0732330	4	24 bytes...	CS0 (0x00)	Not-ECT (...)	32	408 (0x19...	0x00 (Res... 0	0	1	IGMP (2)	0x0000	10
Loopb...	1290.5731080	4	24 bytes...	CS0 (0x00)	Not-ECT (...)	32	409 (0x19...	0x00 (Res... 0	0	1	IGMP (2)	0x0000	10
Loopb...	1292.5325150	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	414 (0x19...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1292.5327630	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	161	10492 (0x...	0x00 (Res... 0	0	1	UDP (17)	0x0000	10
Loopb...	1293.5730060	4	24 bytes...	CS0 (0x00)	Not-ECT (...)	32	416 (0x1A...	0x00 (Res... 0	0	1	IGMP (2)	0x0000	10
Loopb...	1295.0731470	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	172	417 (0x1A...	0x00 (Res... 0	0	128	ICMP (1)	0x0000	10
Loopb...	1295.0732330	4	20 bytes...	CS0 (0x00)	Not-ECT (...)	172	418 (0x1A...	0x00 (Res... 0	0	128	ICMP (1)	0x0000	10

#### Note

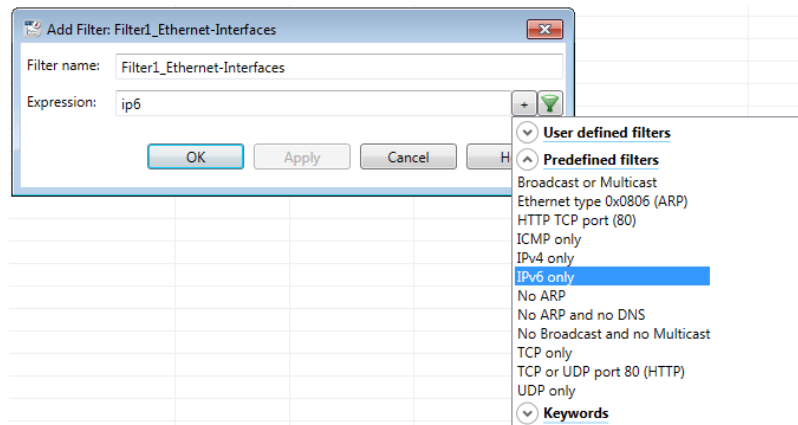
Keep in mind the important information when monitoring and logging Ethernet traffic with ControlDesk.

Refer to [Important Information when Monitoring and Logging Ethernet Traffic](#) ([ControlDesk Introduction and Overview](#)).

For details on monitoring bus communication, refer to [Basics on Monitoring, Logging, and Replaying Bus Communication](#) ([ControlDesk Bus Navigator](#)).

### Ethernet capture filter

The Bus Navigator lets you add a filter to specify the Ethernet packets to be captured. Filter expressions have to be defined according to the Berkeley Packet Filter (BPF) syntax.



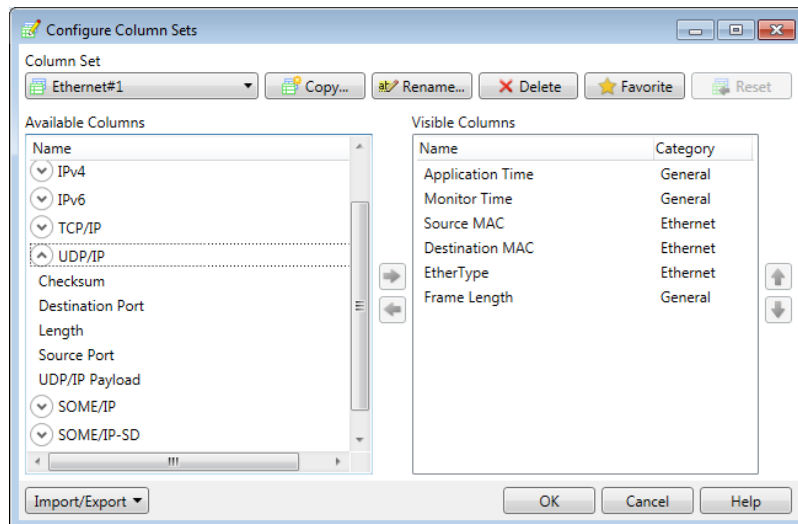
Refer to [Add/Edit Filter Dialog \(Ethernet\)](#) ( ControlDesk Bus Navigator).

### Monitoring list: Column configuration via column sets

The Bus Navigator now lets you configure the columns in a monitoring list via column sets. Column configuration is especially useful for displaying protocol-specific properties if you monitor Ethernet communication.

You can use predefined column sets or specify your own ones. You can export and import the user-defined column sets to/from a CSET file.


The following illustration shows the **Configure Column Sets** dialog for Ethernet monitoring as an example:



Refer to [Configure Column Sets Dialog \(Monitoring List\)](#) ( ControlDesk Bus Navigator).

### Logging bus statistics

The Bus Navigator now lets you save/load bus statistics data (including the bus statistics history) to/from a log file in CSV or ASC format.


For instructions, refer to [How to Display Bus Statistics](#) ( [ControlDesk Bus Navigator](#)).

---

**AUTOSAR/FIBEX file import:  
Support of multicluster files**

AUTOSAR and FIBEX files may contain the description of multiple clusters.

When you add such a variable description file to a bus monitoring device, ControlDesk 6.2 now lets you select the cluster to be imported.

For instructions, refer to [How to Add a Variable Description to a Platform/Device](#) ( [ControlDesk Variable Management](#)).

---

**AUTOSAR file import:  
Support of AUTOSAR 4.3.0**

ControlDesk now also supports AUTOSAR system template version 4.3.0 in connection with the following devices:

- CAN Bus Monitoring Device
- FlexRay Bus Monitoring Device
- LIN Bus Monitoring Device

Refer to [Variable Descriptions Supported by ControlDesk](#) ( [ControlDesk Variable Management](#)).

---

**CAN Bus Monitoring device:  
Displaying all signals of a  
CAN PDU in a Bus instrument  
(RX type)**

If you use a CAN Bus Monitoring device, the Bus Navigator now lets you specify to display each signal of an RX PDU or multiplexed RX PDU in a bus instrument (RX type), regardless of whether the signal is actually sent or received by an ECU.

Refer to [Bus Navigator Page](#) ( [ControlDesk Bus Navigator](#)).

---

**Bus Instrument (TX Type for  
CAN): Enhancement for Bus  
Manager applications**

**New instrument region: PDU Cyclic Timing Control** If the *PDU Cyclic Timing Control* bus simulation feature is configured for the application, the instrument displays the period and offset settings. You can also specify substitute values to be transmitted.

For reference information on the Bus Instrument (TX Type), refer to [Bus Instrument \(TX Type for CAN\)](#) ( [ControlDesk Bus Navigator](#))

---

**Bus Instrument (Inspection  
Layout Type for CAN and LIN):  
Enhancement for Bus  
Manager applications**

**New instrument region: PDU RX Status and Time** If the *PDU RX Status* bus simulation feature is configured for the application, the instrument displays the status and time settings.

For reference information on the Bus Instrument (Inspection Type), refer to:

- [Bus Instrument \(Inspection Type for CAN\)](#) ( [ControlDesk Bus Navigator](#))
- [Bus Instrument \(Inspection Type for LIN\)](#) ( [ControlDesk Bus Navigator](#))

## Related topics

### Basics

[Important Information when Monitoring and Logging Ethernet Traffic](#)  
[\(📖 ControlDesk Introduction and Overview\)](#)

## New Signal Editor Features (ControlDesk 6.2)

### Using custom columns and custom properties

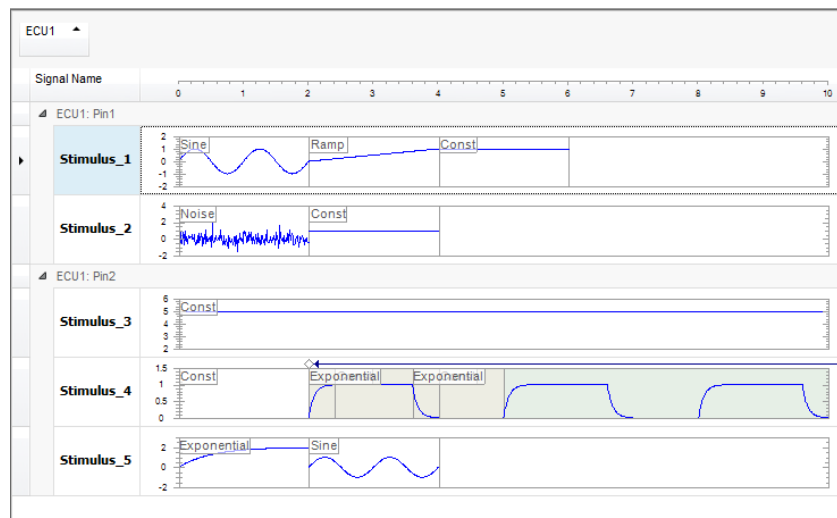
You can now specify one or more additional custom columns in the column headers of signal description sets. Refer to [Create Column](#) (📖 [ControlDesk Signal Editor](#)).

Using custom columns in the header of a signal description set lets you specify your own custom properties for the single signals of the set.


The custom columns and custom properties are stored as specific custom properties in the related STZ file of the signal description set. You can reuse this information when carrying out automated tests at later stages of the development process.

The custom columns also let you group and filter signals in ControlDesk's working area for a better overview. Refer to [Show Group By Box](#) (📖 [ControlDesk Signal Editor](#)).

The following illustration shows a signal description set with five signals as an example. The ECU1 custom column allows you to group the signals.



---

**Highlighting mapped signals** For each signal of a signal generator that is opened in the working area, you can highlight the related variable mapping in the **Signal Mapping**. Refer to [Select Mapping](#) ( [ControlDesk Signal Editor](#)).


---

**Relative MF4 file path for backing up projects with data file segments**

Up to and including version 6.1, ControlDesk stored the *absolute path to the MF4 file* referenced by a data file segment. When you transferred a project backup to another PC, the MF4 file path of the data file segment had to be reconfigured if the MF4 file was located in a user-specific folder such as the default project root folder.


As of version 6.2, ControlDesk now also stores the *relative path* (in addition to the absolute path) to the MF4 file referenced by a data file segment if the file is located in the active ControlDesk project.

When you transfer a project backup to another PC and if a relative path to the MF4 file is available, i.e., if the file is located in the active ControlDesk project, ControlDesk uses the relative path instead of the absolute one. As a result, the MF4 file path of the data file segment does not have to be reconfigured. The absolute file path is changed according to the relative path.

For more information on the segment and its properties, refer to [Data File \(Segment\)](#) ( [ControlDesk Signal Editor](#)).

---

**Specifying multiline comments**

You can edit multiline comments for signals and segments. Refer to [Comment Property](#) ( [ControlDesk Signal Editor](#)).

## Further Enhancements and Changes with ControlDesk (ControlDesk 6.2)

---

**New licensing for dSPACE products**

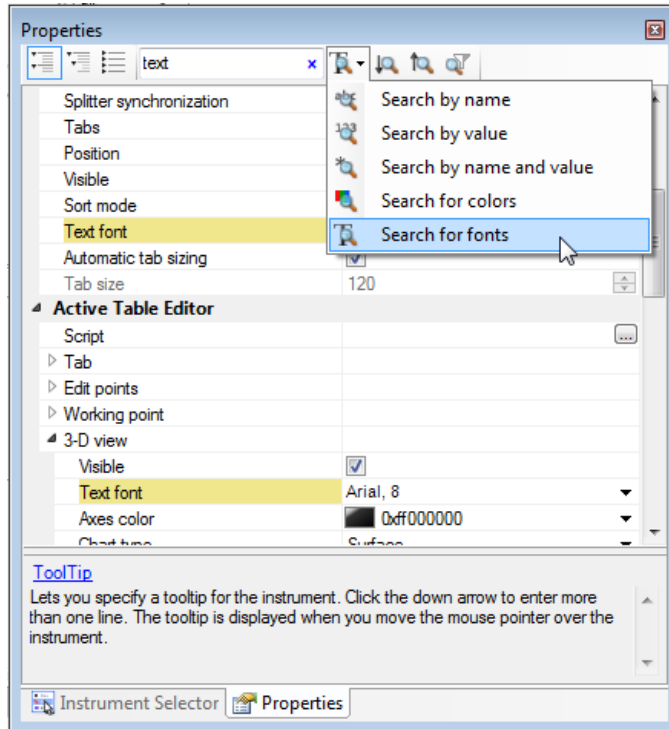
With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

---

**Searching for color and font items in the Properties controlbar**

ControlDesk's **Properties** controlbar now lets you search for color and font items. Search results are highlighted.

The following illustration shows an example:

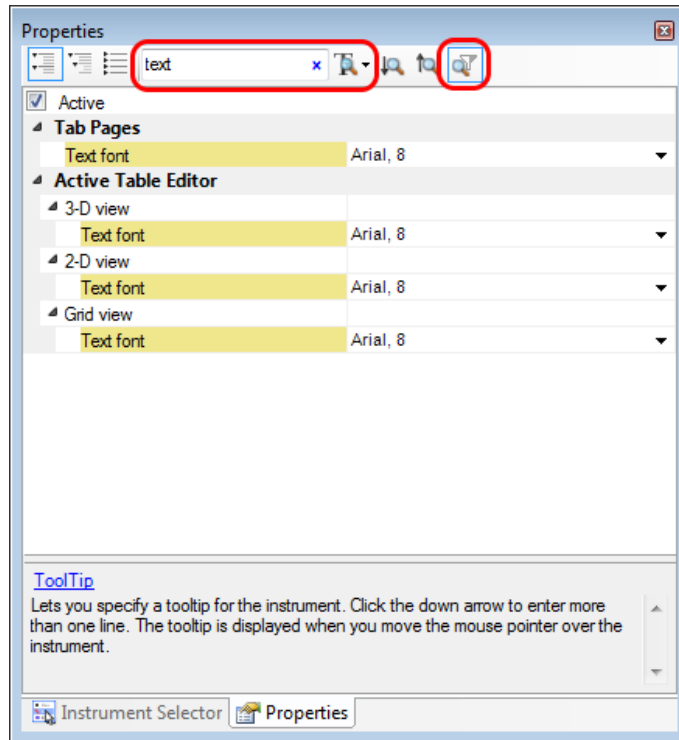


Refer to [Properties \(Controlbar\)](#) ( [ControlDesk User Interface Handling](#)).

### Filtering items in the Properties controlbar

ControlDesk's **Properties** controlbar now lets you filter the controlbar's content by search results. All the matching items including their parent objects are displayed.

The following illustration shows an example:



Refer to [Properties \(Controlbar\)](#) ([ControlDesk User Interface Handling](#)).



# Migrating to ControlDesk 6.2

## Where to go from here

## Information in this section

<a href="#">Discontinuations in ControlDesk</a> .....	113
Gives you an overview of the discontinuations in ControlDesk.	
<a href="#">Migrating to ControlDesk 6.2</a> .....	114
To migrate from ControlDesk 6.1 to ControlDesk 6.2 and reuse existing experiments, you might have to carry out the following migration steps.	

## Discontinuations in ControlDesk

## Where to go from here

## Information in this topic

<a href="#">Discontinuations as of ControlDesk 6.2</a> .....	113
<a href="#">Global platforms/devices</a> .....	113
<a href="#">Loader version</a> .....	113
<a href="#">PCAN tools no longer part of the setup</a> .....	113
<a href="#">Discontinuations for ControlDesk as of dSPACE Release 2018-A</a> .....	114
<a href="#">Third-party PC-based interfaces</a> .....	114

## Discontinuations as of ControlDesk 6.2

**Global platforms/devices** As of ControlDesk 6.2, you can no longer specify a platform/device as a *project-global platform/device*.

For migration aspects, refer to [Global platforms/devices: Discontinuation and migration](#) on page 115.

**Loader version** You can perform the following platform management tasks with ControlDesk *even without a valid license*:

- Registering dSPACE real-time hardware
- Loading, starting, and stopping applications on dSPACE real-time hardware
- Managing the firmware of dSPACE real-time hardware

Up to and including ControlDesk 6.1, performing these tasks without a valid license required that you install **ControlDesk - Loader Version**.

As of ControlDesk 6.2, the installation of **ControlDesk - Loader Version** is no longer available. You can perform these tasks by simply working with ControlDesk without a valid license.

**PCAN tools no longer part of the setup** As of ControlDesk 6.2, the PCAN tools from PEAK-System Technik GmbH are no longer part of the dSPACE Release setup.

### Discontinuations for ControlDesk as of dSPACE Release 2018-A

**Third-party PC-based interfaces** As of dSPACE Release 2018-A, ControlDesk will no longer support some third-party PC-based interfaces. The following table lists the interfaces for which support will be discontinued:

Supplier	Interface
<b>CAN<sup>1)</sup></b>	
Vector Informatik	<ul style="list-style-type: none"> <li>▪ CANcardXL</li> <li>▪ CANcardXLe</li> </ul>
Kvaser	<ul style="list-style-type: none"> <li>▪ LAPcan</li> <li>▪ LAPcan II</li> </ul>
Eberspächer Electronics	<ul style="list-style-type: none"> <li>▪ FlexCard Cyclone II</li> <li>▪ FlexCard Cyclone II SE</li> <li>▪ FlexCard USB</li> </ul>
<b>LIN<sup>2)</sup></b>	
Vector Informatik	<ul style="list-style-type: none"> <li>▪ CANcardXL</li> <li>▪ CANcardXLe</li> </ul>
Kvaser	<ul style="list-style-type: none"> <li>▪ LAPcan</li> <li>▪ LAPcan II</li> </ul>
<b>FlexRay<sup>3)</sup></b>	
Eberspächer Electronics	<ul style="list-style-type: none"> <li>▪ FlexCard Cyclone II</li> <li>▪ FlexCard Cyclone II SE</li> <li>▪ FlexCard USB</li> </ul>

<sup>1)</sup> For a list of CAN interfaces supported by ControlDesk, refer to [Supported CAN Interfaces](#) ([ControlDesk Platform Management](#)).

<sup>2)</sup> For a list of LIN interfaces supported by ControlDesk, refer to [Supported LIN Interfaces](#) ([ControlDesk Platform Management](#)).

<sup>3)</sup> For a list of FlexRay interfaces supported by ControlDesk, refer to [Supported FlexRay Interfaces](#) ([ControlDesk Platform Management](#)).

## Migrating to ControlDesk 6.2

### Introduction

To migrate from ControlDesk 6.1 to ControlDesk 6.2 and reuse existing experiments, you might have to carry out the following migration steps.

#### Note

To migrate to ControlDesk 6.2 from versions earlier than 6.1, you might also have to perform the migration steps of the intervening ControlDesk versions.

**Where to go from here****Information in this topic**

Global platforms/devices: Discontinuation and migration .....	115
Automatic migration .....	115
Manual migration .....	115
Migrating automation scripts .....	116
dSPACE failure simulation hardware controlled via CAN: Migrating to dSPACE CAN	
API 2.0 .....	116
Sampling period of event-based rasters .....	117
Migrating variable connections due to TRC file changes for Bus Manager elements .....	117
Tool automation changes .....	117
Change to the IXaActiveProject interface .....	117
Change to the IBnCANCommunicationChannel and IBnLINCommunicationChannel interfaces .....	118
Change to the VariableDescriptionRemoving event .....	118
Change to the IXaMeasurementRaster interface .....	118
Migrating from prior ControlDesk versions .....	119

**Global platforms/devices: Discontinuation and migration**

As of ControlDesk 6.2, you can no longer specify a platform/device as a *project-global platform/device*.

**Automatic migration** When you open an experiment with global platforms/devices in ControlDesk 6.2 or later, ControlDesk automatically migrates the project and the contained experiments.

In the first experiment that contains global platforms/devices, ControlDesk performs the following steps during experiment migration:

- ControlDesk changes global platforms/devices *in the first experiment* to experiment-specific platforms/devices.
- ControlDesk assigns variable descriptions and data sets that are originally assigned to global platforms/devices *in the first experiment* to the related experiment-specific platforms/devices.

**Manual migration** In all the other experiments that originally contained global platforms/devices, you have to perform the following manual migration steps after ControlDesk's experiment migration:

- Add new platforms/devices *to all the other experiments* that originally contained global platforms/devices.

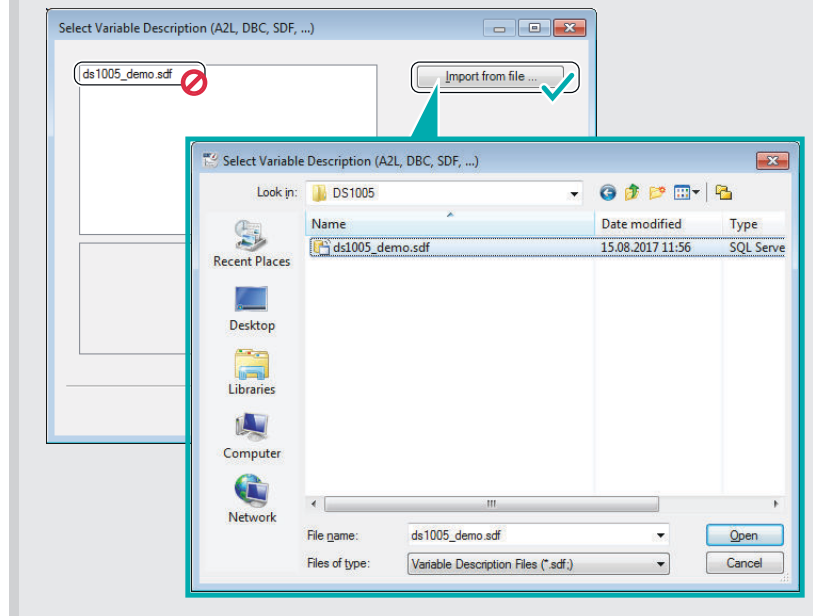
Refer to [How to Add a Platform/Device to an Experiment](#) ( [ControlDesk Platform Management](#)).

- Assign the variable descriptions and data sets originally assigned to global platforms/devices *in all the other experiments* to the related experiment-specific platforms/devices.

**Note**

When you select a variable description that is already assigned to a platform/device *in the first experiment*, you cannot reload and replace the variable description since it is used in other experiments.

Instead of selecting a variable description that is already assigned, *reimport the variable description*, as shown in the following illustration.



Refer to [How to Add a Variable Description to a Platform/Device](#) ([ControlDesk Variable Management](#)).

**Migrating automation scripts** You may have to migrate your automation scripts.

For information on related changes to ControlDesk's automation interface, refer to [Change to the IXaActiveProject interface](#) on page 117.

### dSPACE failure simulation hardware controlled via CAN: Migrating to dSPACE CAN API 2.0

Some dSPACE failure simulation hardware can be controlled by CAN interfaces. To access these CAN interfaces, ControlDesk uses *dSPACE CAN API*.

As of ControlDesk 6.2, when you configure the EESPort for dSPACE failure simulation hardware controlled by CAN interfaces, you can select one of the following versions of dSPACE CAN API:

- dSPACE CAN API 1.0  
dSPACE CAN API 1.0 is supported for compatibility reasons only. Support for dSPACE CAN API 1.0 will be discontinued in the future.
- dSPACE CAN API 2.0  
dSPACE CAN API 2.0 was introduced with dSPACE Release 2016-B. It is the successor of dSPACE CAN API 1.0, includes all previous features, and additionally supports CAN FD. Unlike dSPACE CAN API 1.0, dSPACE CAN API 2.0 will be developed further.

You are therefore recommended to migrate the EESPort configuration to use dSPACE CAN API 2.0.

For reference information, refer to [EESPort - Configuration Properties](#) ([📖 ControlDesk Electrical Error Simulation via XIL API EESPort](#)).

---

### Sampling period of event-based rasters

In ControlDesk 6.2, the value of the sampling period of event-based rasters as displayed in ControlDesk's **Properties** controlbar and as returned by using ControlDesk's automation interface has been changed.

As of ControlDesk 6.2:

- The sampling period value of event-based rasters as displayed in ControlDesk's **Properties** controlbar is "-1 s".
- The sampling period value of event-based rasters as returned by using ControlDesk's automation interface is "-1.0".

Up to and including ControlDesk 6.1, the behavior was as follows:

- Sampling period value of event-based rasters as displayed in ControlDesk's **Properties** controlbar:
  - The sampling period value of event-based rasters *on multiprocessor and multi-core platforms* as displayed in ControlDesk's **Properties** controlbar was "0 s".
  - The sampling period value of event-based rasters *on devices and on all other platforms* as displayed in ControlDesk's **Properties** controlbar was "-1 s".
- Sampling period value of event-based rasters as returned by using ControlDesk's automation interface:
  - The sampling period value of event-based rasters *on multiprocessor and multi-core platforms* as returned by ControlDesk's automation interface was "0.0".
  - The sampling period value of event-based rasters *on devices and on all other platforms* as returned by ControlDesk's automation interface was "-1.0".

---

### Migrating variable connections due to TRC file changes for Bus Manager elements

The paths of Bus Manager elements in the TRC file have changed from dSPACE Release 2017-A to dSPACE Release 2017-B. As a consequence, when you build an application including bus communication configured with the Bus Manager of dSPACE Release 2017-B, you might have to adapt the ControlDesk experiments that use the generated TRC file (e.g., generate new instrument layouts and adapt automation scripts).

---

### Tool automation changes

**Change to the IXaActiveProject interface** As of ControlDesk 6.2, you can no longer specify a platform/device as a *project-global platform/device*.

For this reason, the **Platforms** property of the **ActiveProject / IXaActiveProject <<Interface>>** has been removed.

Refer to [ActiveProject / IXaActiveProject <<Interface>>](#) ([📖 ControlDesk Automation](#)).

**Change to the IBnCANCommunicationChannel and****IBnLINCommunicationChannel interfaces**

In ControlDesk 6.2, the return value of the `Name` property of the following interfaces was changed:

- `CANCommunicationChannel / IBnCANCommunicationChannel <<Interface>>`
- `LINCommunicationChannel / IBnLINCommunicationChannel <<Interface>>`

As of ControlDesk 6.2, the `Name` property of these interfaces returns the channel's `ShortName`. Up to and including ControlDesk 6.1, an internal ID was returned by the `Name` property.

Refer to:

- [CANCommunicationChannel / IBnCANCommunicationChannel <<Interface>>](#) (📖 ControlDesk Automation)
- [LINCommunicationChannel / IBnLINCommunicationChannel <<Interface>>](#) (📖 ControlDesk Automation)

**Change to the VariableDescriptionRemoving event**

In ControlDesk 6.2, the behavior of the `VariableDescriptionRemoving` event of the `VariablesManagementEvents / IXaVariablesManagementEvents <<EventInterface>>` interface was changed.

The `VariableDescriptionRemoving` event is triggered when a variable description is removed from a platform/device. This is possible only for inactive variable descriptions.

A variable description can be referenced *in only one experiment* or *in multiple experiments* of a project:

- When you remove a variable description that is *referenced in only one experiment*, the variable description is removed from the platform/device *and from the project*.
- When you remove a variable description that is *referenced in multiple experiments*, the variable description is removed from the platform/device, *but not from the project*.

The behavior of the `VariableDescriptionRemoving` event has been changed as follows:

- Up to and including ControlDesk 6.1, the `VariableDescriptionRemoving` event was triggered as in the two cases described above, i.e., even if the variable description was removed from the platform/device, but not from the project.
- As of ControlDesk 6.2, the `VariableDescriptionRemoving` event is triggered if the variable description is referenced in only one experiment, i.e., if the variable description is removed from the platform/device and from the project.

Refer to [VariablesManagementEvents / IXaVariablesManagementEvents <<EventInterface>>](#) (📖 ControlDesk Automation).

**Change to the IXaMeasurementRaster interface**

In ControlDesk 6.2, the return value of the `SamplingPeriod` property of the `MeasurementRaster / IXaMeasurementRaster <<Interface>>` interface has been changed; for details on the change, refer to [Sampling period of event-based rasters](#) on page 117.

For details on the interface, refer to [MeasurementRaster / IXaMeasurementRaster <<Interface>>](#) ([📖 ControlDesk Automation](#)).

---

### Migrating from prior ControlDesk versions

To migrate from prior ControlDesk versions and reuse existing experiments, you might have to carry out additional migration steps. For more information on the migration steps, refer to [Migrating from Prior Versions of ControlDesk](#) ([📖 ControlDesk Introduction and Overview](#)).

---

### Related topics

#### Basics

[Basics on Migrating from Prior Versions of ControlDesk](#) ([📖 ControlDesk Introduction and Overview](#))





# DCI Configuration Tool

## New Features of the DCI Configuration Tool 3.8

---

### Improved A2L file adaptation

The DCI Configuration Tool features improvements related to the adaptation of an existing A2L file that is used with a DCI-GSI2.

---

### Firmware versions for DCI-GSI1 and DCI-GSI2 interfaces

The following firmware versions for the DCI-GSI1 and DCI-GSI2 interfaces are delivered with the DCI Configuration Tool 3.8:

- DCI-GSI1 firmware version 1.6.8
- DCI-GSI2 firmware version 1.4.9

#### Note

The firmware version delivered with the DCI Configuration Tool is not always the latest firmware version available. If you encounter problems, contact dSPACE Support to check if a later firmware version is available.



# dSPACE CAN API Package

## New Features of dSPACE CAN API Package 3.0.2

---

### **Checking the availability of a specific CAN channel**

The dSPACE CAN API 2.0 included in dSPACE CAN API Package 3.0.2 provides a function to check the availability of a specific CAN channel.

Refer to [DSCAN\\_IsChannelAvailable](#) ( [dSPACE CAN API 2.0 C Reference](#)).



# dSPACE ECU Flash Programming Tool

## New Features of the dSPACE ECU Flash Programming Tool 2.4

---

### **New licensing for dSPACE products**

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### **Managing file associations of project files and ECU data files**

The dSPACE ECU Flash Programming Tool provides a new dialog to select file name extensions that are associated with the tool. The dSPACE ECU Flash Programming Tool is used for opening files of the selected file types.

Refer to [Set File Associations Dialog](#) ( [ECU Flash Programming](#)).



# dSPACE FlexRay Configuration Package

## New Features of dSPACE FlexRay Configuration Package 4.0

---

### **New licensing for dSPACE products**

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

---

### **New supported platform**

The FlexRay Configuration Package supports SCALEXIO systems with a DS6311 FlexRay Board. The DS6311 FlexRay Board provides four FlexRay communication controllers each with a FlexRay channel A and a FlexRay channel B.





# dSPACE XIL API .NET

## New Features of dSPACE XIL API .NET 2017-B

---

### **New licensing for dSPACE products**

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### **Enhanced MAPort functionality**

**Enhancement to the SignalGenerator** Stimulus signals now support the scaling parameter in the variable description file.

The scaling is enabled by default. To work with the stimulus behavior as it was in previous versions of dSPACE XIL API .NET, you must disable scaling via the `EnableSignalGeneratorScaling` setting in the MAPort configuration file. For more information, refer to [MAPort Configuration](#) ([📖 dSPACE XIL API Reference](#)).

### **Enhanced EESPort functionality**

**Support of dSPACE CAN API 2.0** If you use dSPACE FIU boards that are connected to the simulator hardware via CAN bus you can now configure the dSPACE CAN API version 2.0 as an interface in the EESPort configuration file.

**Custom properties** You can now specify custom properties in the error configuration file. This lets you add additional data to certain EESPort objects that you can use during run time or for result evaluation, for example.

For more information, refer to [Implementing an EESPort Client Application](#) ([📖 dSPACE XIL API Implementation Guide](#)).



# ECU Interface Manager

## Where to go from here

## Information in this section

<a href="#">New Features of ECU Interface Manager 2.2</a> .....	131
An overview of the new features of ECU Interface Manager 2.2.	
<a href="#">Migrating to ECU Interface Manager 2.2</a> .....	132
Information on how to migrate to ECU Interface Manager 2.2.	

## New Features of ECU Interface Manager 2.2


### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### ECU calibration page handling via XCP

The ECU Interface Manager 2.2 now lets you configure the access to ECU calibration pages in connection with ConfigurationDesk/SCALEXIO.

ECU calibration page handling lets you switch the active ECU calibration page, for example.

For details, refer to [Preparing ECU Calibration Page Handling](#) ( [ECU Interface Manager Manual](#)).

### Data access configuration for ECU variables

The ECU Interface Manager 2.2 now lets you configure read/write access to ECU variables. Variable access is based on the asynchronous upload/download of


measurement variables and characteristics. The ECU Interface Manager lets you access these ECU variables in connection with ConfigurationDesk/SCALEXIO.

For details, refer to [Basics on Configuring Data Accesses](#) ( ECU Interface Manager Manual).

---

## Related topics

### Basics

- [Basics on Configuring Data Accesses](#) ( ECU Interface Manager Manual)
- [Preparing ECU Calibration Page Handling](#) ( ECU Interface Manager Manual)

---

## Migrating to ECU Interface Manager 2.2

### Automatic migration of projects

You can reuse projects in ECU Interface Manager 2.2 if the projects were last saved with one of the following ECU Interface Manager versions:

- ECU Interface Manager 2.0 p1
- ECU Interface Manager 2.1

When you open the projects in ECU Interface Manager 2.2, they are migrated automatically.

#### Note

In ECU Interface Manager 2.2, you cannot reuse projects that were last saved with ECU Interface Manager 2.0 or earlier.

---

### Automation change

In ECU Interface Manager 2.2, the `FromFunctionAccessConfiguration` attribute of the `Function` parameter of the `Delete` method was removed.

As of ECU Interface Manager 2.2, to delete a function access from the **Access Configuration** pane, you can use the `AccessConfiguration` parameter of the `Delete` method.

Refer to [Delete](#) ( ECU Interface Manager Manual).

# Model Compare

## Where to go from here

## Information in this section

<a href="#">New Features of Model Compare 2.8</a> .....	133
<a href="#">Migration to Model Compare 2.8</a> .....	135

## New Features of Model Compare 2.8

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### Conflict detection

A three-way analysis of the reference, the comparison and the common ancestor models not only detects the differences between the comparison and the reference model. If model elements or properties in the common ancestor model are modified differently in the reference model and in the comparison model they differ in all three models. These modifications include additions and removals. In the **Model Navigator**, the related model elements are now marked as conflicts with an exclamation point. New commands let you skip or filter the model elements with detected conflicts.

#### Related documentation

- [Model Navigator](#) ([Model Compare Reference](#))
- [Previous/Next Conflict](#) ([Model Compare Reference](#))

- [Comfort Copy to Right](#) ([📖 Model Compare Reference](#))
- [Comfort Copy to Left](#) ([📖 Model Compare Reference](#))

---

### Advanced merging of model elements (Comfort Copy commands)

In addition to the known commands to copy model elements and properties from the reference model to the comparison model and vice versa, Model Compare introduces **Comfort Copy** commands.

A three-way analysis not only detects the differences between the three models but also conflicts where the comparison model and the reference model are modified differently. Via the provided **Comfort Copy** commands, you can copy all modified but non-conflicting elements at all levels of the selected subsystems into the target model in one step. Modifications of the target model remain unchanged.

#### Related documentation

- [Basics on Merging Models](#) ([📖 Model Compare Guide](#))
- [How to Merge a Node's Non-Conflicting Model Elements at Once](#) ([📖 Model Compare Guide](#))
- [Comfort Copy to Right](#) ([📖 Model Compare Reference](#))
- [Comfort Copy to Left](#) ([📖 Model Compare Reference](#))

---

### Additional filters for blocks, lines, and annotations

To support your work with **Comfort Copy** commands and detected conflicts, Model Compare provides additional filters to display only currently relevant model elements.

#### Related documentation

- [Display Page](#) ([📖 Model Compare Reference](#))

---

### Enhanced display filters

The definition of some basic filters are enhanced to make their results more significant.

#### Related documentation

- [Basic Filters Page](#) ([📖 Model Compare Reference](#))

---

### Convenient filter specification

You can specify to filter a block type or a property type via the context menu of a related element in the **Model Navigator** or the **Property Inspector**.

#### Related documentation

- [Add to Block Type Filter](#) ([📖 Model Compare Reference](#))
- [Add to Property Type Filter](#) ([📖 Model Compare Reference](#))

---

### Support of commented out blocks

Model Compare supports commented out and commented through blocks and subsystems during the model comparison.

#### Related documentation

- [Basic Filters Page](#) ([📖 Model Compare Reference](#))

**Improved report generation**

- You can now start the report generation via the toolbar.
- At the start of the report generation, a default report name based on the session and the model name is offered.
- You can now specify to truncate the printing of long, hardly readable values to a maximum length to make reports easier to read.
- You can select to suppress outputs of the TargetLink simulation frame to keep the reported information short and significant.
- Each model screenshot is only added once to a report.

**Related documentation**

- [Create Report](#) ( [Model Compare Reference](#))
- [Report Options Dialog](#) ( [Model Compare Reference](#))

## Migration to Model Compare 2.8

---

**No adaptation necessary**

You can migrate from Model Compare 2.7 to Model Compare 2.8 without adaptations.





# ModelDesk

---

## Where to go from here

## Information in this section

New Features of ModelDesk 4.6 .....	137
Migration to ModelDesk 4.6 .....	139

---

## New Features of ModelDesk 4.6

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

---

### ModelDesk license

ModelDesk requires only one license. This license includes all previous ModelDesk licenses.

---

### Project management

**ASM project** You can create a new ModelDesk project by selecting an ASM demo model. ModelDesk extracts the demo model in a work folder and creates a project and experiment for it.

---

### Parameterizing

**Extended parameter address** The parameter address that is displayed in the **Parameter Details** properties of the parameters is extended so it can be used in MATLAB using copy & paste.

**Restructured view of the parameter set** You can restructure the nodes in the Parameter Set node of the project tree. This feature gives you another view to the parameters of the simulation model. It allows for the following use scenarios, for example:

- You can merge the parameter pages of several main components, for example, engine and drivetrain, under one node.
- You can create nodes for each variant of a main component.
- You can merge the parameter pages of main components and custom blocks under the same node.

The new structure is specified in the simulation model. You insert **ModelDesk Parameter Group** blocks into the subsystems of the ASM. When ModelDesk parses the model, these blocks become the nodes that contain the parameter pages which correspond to the ASM blocks of the subsystem.

---

## Processing

**Improved usability** Now the **Processing** properties of the parameters contain a button for creating and editing function and settings files.

**Extended parameter address** The parameter address that is displayed in the **Parameter Details** properties of the parameters is extended so it can be copied & pasted in MATLAB.

---

## Traffic object management

The properties of the traffic objects are extended so you can model traffic objects that are controlled by the simulation, for example, traffic lights and sign gantries.

**New property** The traffic objects have the new **Dynamic** property. You can enable the property only for smart objects that have one state. When the property is enabled, you can set the state during the simulation.

**New sensor** ModelDesk supports a new traffic sign sensor for detecting traffic objects whose properties comply with the standard settings of the traffic signs for road traffic regulations, which is also used by the OpenDRIVE standard, for example. The sensor supports traffic objects that have multiple states.

The previous **Traffic sign** sensor is renamed to **Traffic sign basic**.

---

## Road generation

**Specifying shapes** When you specify shapes, you can use the mouse to set the tangent angle of the nodes.

**New shape type: Trajectories** You can define a new type of shapes on road elements and junctions: Trajectories. Trajectories specify a path that can be used as an alternative to the preferred lanes on road elements or default junction crossing paths.

**New shape type: Guard rail** You can define a new type of shapes on road elements and junctions: Guard rail. Guard rails are visualized in MotionDesk and can be evaluated in the simulation. Guard rails were already part of the Highway scenery. Elements of a scenery are not evaluated in the simulations but the Highway guard rails can now be converted to shape guard rails.

**Export and import of shapes** You can export and import the specified values of a shape in the MAT format.

**Extended route** When you specify a route for the ASM vehicle or the traffic fellows, you can now choose between preferred lanes, default junction crossing paths, and existing trajectories for each section of the route.

**Scene synchronization** Now you can synchronize the traffic objects or the scenery only.

**Tool automation** The object model of the tool automation was adapted to support the new features.

---

#### Maneuver creation

**Steering mode** You can specify a torque-based steering mode.

---

#### Traffic scenario creation

**Improved view** The representation of fellows and global user signals can be collapsed or expanded.

**Filtering** You can specify a filter to reduce the number of fellows and global user signals that are visible in the working view.

---

#### Plotting

**Downsampling** You can specify a downsampling factor for the plotting. Downsampling improves the performance and extends the recording time.

**Signal buffer size** You can specify a signal buffer size to limit the memory consumption of the measurements.

---

#### Related topics

##### Basics

[New Features of ASM Environment Blockset 4.7](#) ..... 66

## Migration to ModelDesk 4.6

---

#### Tool automation for plotting

As of ModelDesk 4.4, ModelDesk has new plotters and the tool automation for plotting was changed. To reuse scripts for plotting, you must adapt scripts written for ModelDesk 4.3 and earlier.

---

#### Triggering of plots

As of ModelDesk 4.6, the plotting is triggered by the simulation. Before that, ModelDesk triggered the plotting. Normally, the plots are equal, but may differ in some cases.

**Tip**

To compare measurements, it is useful to use the XY Plotter and use the maneuver time as a signal for the x-axis.

# Model Interface Package for Simulink

## Where to go from here

## Information in this section

New Features of the Model Interface Package for Simulink 3.5 ..... 141

Migration Aspects of the Model Interface Package for Simulink ..... 144

## New Features of the Model Interface Package for Simulink 3.5

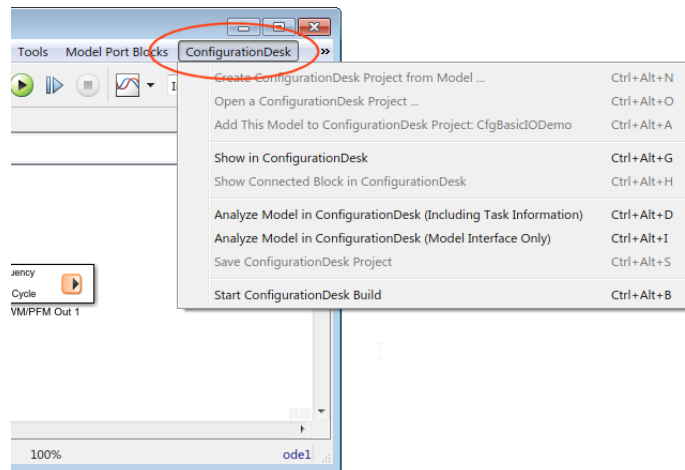
### Improved connection to ConfigurationDesk

The Model Interface Package for Simulink now provides the following features that improve the connection of Simulink behavior models to ConfigurationDesk:

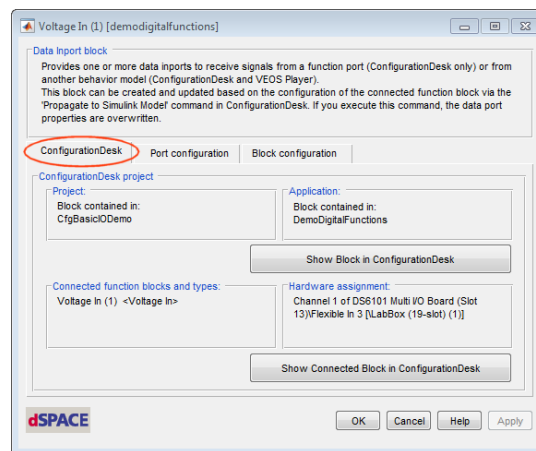
- You can now create a ConfigurationDesk project directly from the Simulink behavior model, or you can add the Simulink model directly to the active ConfigurationDesk application. The Simulink model is then preconfigured for use in ConfigurationDesk automatically, if required.
- You can now switch from a selected model port block in Simulink directly to its representative in ConfigurationDesk, or to the function block to which the representative is mapped.
- Important ConfigurationDesk features, such as starting a ConfigurationDesk build process or analyzing the Simulink model, are now accessible directly in the Simulink model.
- The block dialogs of the blocks provided by the Model Port Block Library have been revised and unified.

The new features can be accessed as follows:

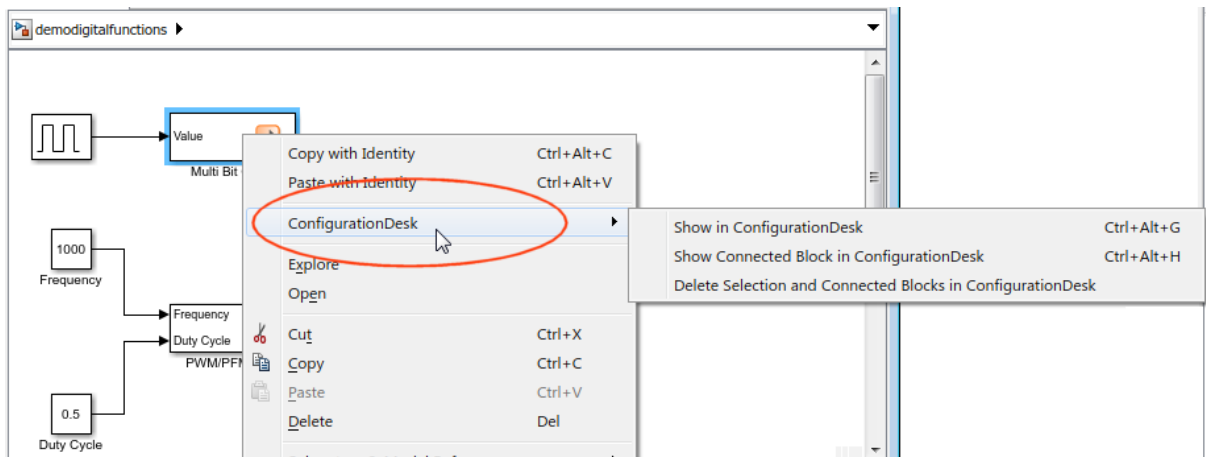
- Via the **ConfigurationDesk** menu in the menu bar of your Simulink behavior model. The **ConfigurationDesk** menu provides access to commands for the remote access of ConfigurationDesk:



- Via the **ConfigurationDesk** page in the dialogs of the blocks of the Model Port Block Library. The **ConfigurationDesk** page lets you view information about the selected model port block in the related ConfigurationDesk project and application:



- Via the **ConfigurationDesk** context menu of selected model port blocks. The **ConfigurationDesk** context menu lets you switch to related model port blocks or to the connected function blocks in ConfigurationDesk. Additionally, you can delete selected model port blocks including the related signal chains in ConfigurationDesk:



### Benefits of creating a ConfigurationDesk project from a Simulink model

Suppose you specified a default hardware in ConfigurationDesk, and you opened a new empty model in Simulink. With only a few clicks, you can now create a ConfigurationDesk project and application via the **Create ConfigurationDesk Project From Model** command. The new project has the following specifics:

- The new ConfigurationDesk application contains a Simulink model.
- ConfigurationDesk shows a list of the function blocks that are available for the specified default hardware.
- There are no conflicts shown in the **Conflicts Viewer**.

### Benefits of switching directly to a related model port block in ConfigurationDesk

The Model Interface Package for Simulink now provides commands for switching directly to related model port blocks or function blocks in ConfigurationDesk. You can directly view and change the configuration of the function block. If necessary, you can propagate changes in the function block configuration back to the Simulink behavior model to resynchronize the ConfigurationDesk model interface and the Simulink model interface.

#### Tip

The **Ctrl+Alt+G** and **Ctrl+Alt+H** shortcuts are available in Simulink and ConfigurationDesk to make switching between the model port blocks of the two tools easier.

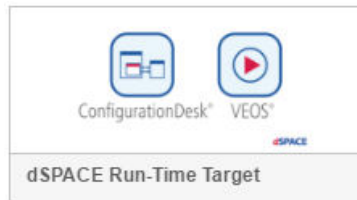
#### Note

The **ConfigurationDesk** menu, the **ConfigurationDesk** page of the model port block dialogs, and the **ConfigurationDesk** context menu are available if ConfigurationDesk is installed.

For a detailed description, refer to [Remote Access to ConfigurationDesk](#) ([Model Interface Package for Simulink - Modeling Guide](#)).

### New model template for the dSPACE Run-Time Target

The Model Interface Package for Simulink provides a template for Simulink models that is preconfigured for the use in ConfigurationDesk or VEOS Player. You can create new Simulink models based on this template. The models are then preconfigured for the dSPACE Run-Time Target. The template is accessible via the Simulink Start Page:



For details, refer to [Basics on Simulink Implementation Containers](#) ([Model Interface Package for Simulink - Modeling Guide](#)).

### Unsupported new features of MATLAB R2017b

The following new features introduced with MATLAB R2017b are not supported by the Model Interface Package for Simulink:

- Simulink Coder allows you to specify a custom folder for generating code and temporary fragments for simulation. This new option called **CodeGenFolderStructure** is not supported. The same applies to other code generation folder settings such as **CacheFolder**, and **CodeGenFolder**.
- Simulink Coder provides the option **Allow tasks to execute concurrently on target**. The Model Interface Package for Simulink does not support this feature.
- Values of variables or parameter objects stored in a model workspace can now be changed during simulation by using the **Model Explorer Contents** pane. Parameters defined in the model workspace of referenced models and block level references to the parameters do not appear in the generated variable description.

## Migration Aspects of the Model Interface Package for Simulink

### DSMPBLIB menu renamed

The **DSMPBLIB** menu of the Model Port Block Library has been renamed to **Model Port Blocks**.

### Unified block parameters of state data in MATLAB R2017b

Before MATLAB R2017b, for some blocks you had to use different block parameters to programmatically configure the name and the initial value of state data of different blocks. As of MATLAB R2017b, the block parameters are unified as follows:

- To configure the name of state data, **StateName** is used.
- To configure the initial value of state data, **InitialCondition** is used.



The related entries in TRCs files will change accordingly. You will have to adjust layouts or scripts that access these parameters. The following table shows an example:

<b>R2017a:</b>	<b>R2017b:</b>
.../Rate Transition/X0	.../Rate Transition/InitialCondition



# MotionDesk

---

## Where to go from here

## Information in this section

New Features of MotionDesk 4.1 .....	147
Migrating to MotionDesk 4.1 .....	149

---

## New Features of MotionDesk 4.1

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

---

### MotionDesk license

One license is required for using MotionDesk. Other licenses are required to use special features.

---

### Endless ground and sky

You can use a built-in endless ground and an endless sky to build the virtual world for the scene. This is especially useful when you work with very long road networks. It is not necessary to use a ground plate and dome 3-D object for creating the virtual world.

---

### Scene generation

**Guard rails** MotionDesk can generate guard rails that are specified as shape in ModelDesk.

**Traffic lights and sign gantries** MotionDesk can generate traffic lights and sign gantries that are specified as traffic objects in ModelDesk. These traffic objects can change their state controlled by the simulation application.

---

### Advanced lighting mode

The advanced lighting mode improves the lighting effects in the scene to make it look more realistic. The advanced lighting mode is protected by a special license.

**Large number of light sources** In this mode, MotionDesk has the ability to render many lights in a scene without significant hardware requirements.

**Adding light sources to 3-D objects** You can add and delete lights to static and movable 3-D objects. You can specify the light cone and the position relative to the 3-D object. The lights can be controlled by the simulation application.

**Extended 3-D objects library** Light sources were added to several 3-D objects of the dSPACE objects library.

**Examples** The following illustrations show some examples in the advanced lighting mode.



**Tool automation**

The atmospherics can be specified with the tool automation.

## Migrating to MotionDesk 4.1

**Using endless ground plate and horizon**

In MotionDesk 4.0 and earlier, the virtual world of a scene was built using ground plate and dome 3-D objects. If you want to use the endless ground plate and sky, such 3-D objects are obsolete. When you use an old scene, delete these objects before activating the endless ground and sky.

**Using advanced lighting mode**

In advanced lighting mode, the static objects used for domes are not suitable for building the virtual world. Use the endless horizon of the environment instead.

**Migrating 3-D custom objects**

If you want to use 3-D custom objects in VRML2 format that you used in MotionDesk 2.2.1 or earlier, you have to convert the VRML2 files into COLLADA format files. You can convert the files at any time using the **3-D Library Manager**.

**Migrating from MotionDesk version 2.2.1 and earlier**

The current MotionDesk version cannot read old MotionDesk experiments in the MDX file format (used in MotionDesk 2.1.6 and earlier) or scenes stored in ESD format (used in MotionDesk 2.2.1 and earlier). It is therefore not possible to migrate from MotionDesk projects and experiments of these versions.

If you want to use older projects and experiments, you can migrate them by using MotionDesk 3.0 up to MotionDesk 3.6 and then open them in the current MotionDesk version.



# Real-Time Testing

---

## Where to go from here

## Information in this section

<a href="#">New Features of Real-Time Testing 3.3</a> .....	151
<a href="#">Migrating to Real-Time Testing 3.3</a> .....	152

---

## New Features of Real-Time Testing 3.3

### **New licensing for dSPACE products**

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

---

### **New module: watcherlib**

RTT variables can be monitored depending on conditions that can be specified according to the ASAM General Expression Syntax (GES) standard. The watcherlib module provides the necessary class for this.

---

### **Scaling**

Real-Time Testing provides access to the variables of a simulation application. Now, Real-Time Testing can consider the scaling of these variables if this is specified in the TRC variable description file. Real-Time Testing supports only the linear scaling of parameters, signals, and single elements of vectors and matrices. You can enable or disable the scaling support for each variable separately.

---

**Extended `rttlib.dscanapilib` module**

The `rttlib.dscanapilib` module provides two new methods:

- `enableBusStatistics`: To enable or disable the periodic generation of bus statistics messages for a CAN channel.
- `encodeBusStatistics`: To encode CAN bus statistics information from data bytes of a CAN bus statistics message.

---

**C# demos**

Now, the demos that are installed with Real-Time Testing contain a loader function written in C#. The files demonstrate how to manage RTT sequences using the C# programming language.

---

**Related topics****Basics**

[Checking Conditions According to the ASAM GES Standard](#) (📖 Real-Time Testing Guide)

[Read/Write Access to Variables of the Simulation Application](#) (📖 Real-Time Testing Guide)

**Examples**

[Demo Examples of Using Real-Time Testing](#) (📖 Real-Time Testing Guide)

**References**

[rttlib.dscanapilib Module](#) (📖 Real-Time Testing Library Reference)

---

## Migrating to Real-Time Testing 3.3

**Scaling**

Real-Time Testing considers the scaling of variables of a simulation application if this is specified in the TRC variable description file. The following points must be considered when you migrate:

- When you use RTT sequences that access variables for which scaling is defined, its behavior changes. The previous versions of Real-Time Testing ignore the scaling, Real-Time Testing 3.3 considers the scaling by default. You can disable the scaling by modifying the RTT sequences.
- Scaling can only be considered if Real-Time Testing 3.3 is used. For DS1005, DS1006, and MicroAutoBox, Real-Time Testing is integrated in the simulation application when it is built. To use scaling, you have to rebuild the simulation application.



---

**Fixed parameter**

Fixed parameters cannot be modified. The INITONLY flag is set for fixed parameters, i.e., they have a fixed value in a running simulation and can only be modified during the initialization phase of a simulation application.

---

**Incompatible BCG files**

The BCG files generated with Real-Time Testing 2.6 or earlier cannot be used for Real-Time Testing 3.3. You must create the BCG file of the Real-Time Testing sequence again. Refer to [Creating and Starting RTT Sequences in Python Scripts](#) ([📖 Real-Time Testing Guide](#)).



# RTI/RTI-MP and RTLib

## Where to go from here

## Information in this section

New Features of RTI/RTI-MP and RTLib .....	155
Migration Aspects of RTI/RTI-MP and RTLib .....	156

## New Features of RTI/RTI-MP and RTLib

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### New features in RTI/RTI-MP

The following feature in RTI/RTI-MP is introduced with dSPACE Release 2017-B:

- In a structured data type, the parameter names can now be set to names, which are reserved as keyword in a TRC file. For example, you can use *value* or *default* as name.

### Unsupported new features of MATLAB R2017b

The following new features introduced with MATLAB R2017b are not supported by RTI/RTI-MP:

- Simulink Coder allows you to specify a custom folder for generating code and temporary fragments for simulation. This new option called `CodeGenFolderStructure` is not supported. The same applies to other code generation folder options such as `CacheFolder`, and `CodeGenFolder`.

- Simulink Coder provides the option **Allow tasks to execute concurrently on target**. RTI and RTI-MP do not support this feature. However, concurrent execution of tasks can be done by using RTI-MP for running tasks on multiprocessor or multicore systems.
- Values of variables or parameter objects stored in a model workspace can now be changed during simulation by using the **Model Explorer Contents** pane. Parameters defined in the model workspace of referenced models and block level references to the parameters do not appear in the generated variable description.

---

### Installing Texas Instruments compiler

After you installed a TI compiler, you have to configure the installation path. You can call `DsConfigTiEnv` in the **Command Prompt for dSPACE RCP and HIL** to open a dialog for entering the values of the environment variables `TI_ROOT` and `C2X_ROOT`.

For more information, refer to [How to Set the Compiler Path \(DS2210 RTLib Reference\)](#), [How to Set the Compiler Path \(DS2211 RTLib Reference\)](#), [How to Set the Compiler Path \(DS2302 DSP Programming\)](#), or [How to Set the Compiler Path \(DS1103 RTLib Reference\)](#).

## Migration Aspects of RTI/RTI-MP and RTLib

---

### Changes in TRC file generation

With MATLAB R2017b, the names of the state data is harmonized. Now, all blocks use the same parameter names for the states, i.e., **StateName** and **InitialCondition**. Accordingly, a newly generated TRC file contains the changed parameter names.

For example, up to MATLAB R2017a a TRC entry for the Rate Transition block was `.../Rate Transition/X0`. As of MATLAB R2017b, the TRC entry changed to `.../Rate Transition/InitialCondition`.

Layouts or scripts that reference such a state parameter, must be adapted.

---

### Modified features in later MATLAB versions

**Switching to a later MATLAB version** If you install a new MATLAB version, some settings are adopted from previously installed MATLAB versions. To prevent unexpected behavior of your Simulink models when switching to a later MATLAB version or dSPACE Release, always reset the MATLAB and Simulink preferences to their defaults before you start using them.

# RTI Bypass Blockset

## Where to go from here

## Information in this section

<a href="#">New Features of the RTI Bypass Blockset 3.9</a> .....	157
<a href="#">Migrating to RTI Bypass Blockset 3.9</a> .....	157

## New Features of the RTI Bypass Blockset 3.9

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

## Migrating to RTI Bypass Blockset 3.9

### Working with models from earlier RTI Bypass Blockset versions 3.x and 2.x

The current Release contains RTI Bypass Blockset 3.9, which is compatible with earlier blockset versions 3.x and 2.x. However, there are some points to note:

- *Working with models from RTI Bypass Blockset 2.5 or earlier*

Data management was changed in comparison to the prior RTI Bypass Blockset versions. If you have a Simulink model built with RTI Bypass Blockset 2.5 or earlier and you open it with RTI Bypass Blockset 3.9, the old data dictionary file (with the file name extension .dd) is replaced by a new data dictionary file (.vdb) using the information stored in the Setup block. This

happens as soon as you open and close the Setup block dialog by clicking **OK**, or you open the Read, Write, Upload, or Download block dialog and click **Fill Variable Selector** on the Variables page.

If you have a model that was saved with RTI Bypass Blockset 3.8 and want to use it with RTI Bypass Blockset 2.5 or earlier, the model's data dictionary file required for blockset version 2.5 or earlier (file name extension .dd) is created. This happens as soon as you update the A2L files in the Setup block, or you open the Read, Write, Upload, or Download block and click **Fill Variable Selector** on the Variables page. The data dictionary file created under RTI Bypass Blockset 3.9 (.vdb) remains on the disk.

To enable the RTI Bypass Blockset to recreate the data dictionary, the database files specified in the Setup block must be accessible at the specified location and must be unchanged.

- *Working with models from RTI Bypass Blockset 2.6 up to and including RTI Bypass Blockset 3.8*

If you have a Simulink model built with RTI Bypass Blockset 2.6 up to RTI Bypass Blockset 3.8, and you open it with RTI Bypass Blockset 3.9, the old data dictionary file is replaced by a new data dictionary file. However, the new data dictionary file cannot be used in earlier RTI Bypass Blockset versions. If you want to reuse the model with RTI Bypass Blockset 2.6 up to RTI Bypass Blockset 3.8, you have to create a suitable database in the earlier RTI Bypass Blockset version by reimporting the database files (A2L files) specified in the Setup block.

# RTI CAN MultiMessage Blockset

## Where to go from here

## Information in this section

New Features of the RTI CAN MultiMessage Blockset 4.6 .....	159
Migrating to RTI CAN MultiMessage Blockset 4.6 .....	160

## New Features of the RTI CAN MultiMessage Blockset 4.6

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### New supported platform

The RTI LIN MultiMessage Blockset supports SCALEXIO systems with a DS6341 CAN Board. The DS6341 CAN Board provides four CAN/CAN FD channels.

### Support of AUTOSAR System Template 4.3.0

The RTI CAN MultiMessage Blockset supports the AUTOSAR System Template based on AUTOSAR Release 4.3.0 for describing CAN networks.

Refer to [General Settings Page \(RTICANMM MainBlock\)](#) ( [RTI CAN MultiMessage Blockset Reference](#)).

### Support of AUTOSAR E2E protection profiles 05 and 06

The RTI CAN MultiMessage Blockset now also supports end-to-end communication protection (E2E protection) according to the AUTOSAR end-to-end protection profiles 05 and 06.

Refer to [Checksum Definition Page \(RTICANMM MainBlock\)](#) ( [RTI CAN MultiMessage Blockset Reference](#)).

## Migrating to RTI CAN MultiMessage Blockset 4.6

### Changed MEX compiler support

The MEX compiler support of the RTI CAN MultiMessage Blockset has changed. In earlier blockset versions, the Microsoft Windows SDK compiler was required to build MEX functions. With RTI CAN MultiMessage Blockset 4.6, only the following MEX compilers are supported:


- MinGW (GNU Compiler Collection (GCC 4.9.2)): In combination with MATLAB Releases 2016a, 2016b and 2017a.
- MinGW (GNU Compiler Collection (GCC 5.3.0)): In combination with MATLAB Release 2017b.
- Microsoft Visual Studio 2015 Pro: In combination with MATLAB Release 2016a, 2016b, 2017a and 2017b.

### Working with models from earlier RTI CAN MultiMessage Blockset versions

To reuse a model created with an earlier RTI CAN MultiMessage Blockset version, you must update the S-functions for all the RTICANMM blocks and save the model before modifying the CAN configuration.

To create new S-functions for all the RTICANMM blocks in your model in one step, you can perform one of the following actions after opening the model:

- In the MATLAB Command Window, enter `rtimmsu_update('System', gcs)`.  
For more information on the command and its options, enter `help rtimmsu_update` in the MATLAB Command Window.
- Select the **Create S-Function for all CAN Blocks** command from the **Options** menu of the RTICANMM GeneralSetup block.

For more information, refer to [Limitations with RTICANMM](#) ( [RTI CAN MultiMessage Blockset Reference](#)).

### Compiler messages when using code generated by an RTI CAN MultiMessage Blockset version < 4.0

If you use code that was generated by an RTI CAN MultiMessage Blockset version < 4.0, several compiler warning messages that contain the phrase `<<argument of type "can_tp1_canChannel *" is incompatible with parameter of type "DsTCanCh">>` will be displayed during the build process of your simulation model. This is due to a modified data type. These warnings can be ignored and disappear after you use the current blockset version to generate the RTICANMM code again.



**Using existing checksum algorithms**

Checksum algorithms that were originally developed for an application and contain CAN messages cannot be reused for applications that contain CAN FD messages, because CAN FD includes new message types and longer data fields. Existing checksum algorithms can still be used for applications that contain only classic CAN messages. For CAN FD applications, you must adapt the checksum algorithms.



# RTI FPGA Programming Blockset

## Where to go from here

## Information in this section

New Features of the RTI FPGA Programming Blockset 3.4 .....	163
Migrating to RTI FPGA Programming Blockset 3.4 .....	165

## New Features of the RTI FPGA Programming Blockset 3.4

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### Extended Xilinx® support

The RTI FPGA Programming Blockset now supports the following products and versions of the Xilinx design tools:

Xilinx Design Tools Version	MATLAB Version <sup>1)</sup>	Operating System
64-bit version of Vivado 2017.2	64-bit versions of: <ul style="list-style-type: none"> <li>▪ MATLAB R2016a</li> <li>▪ MATLAB R2016b</li> <li>▪ MATLAB R2017a</li> </ul>	All PC operating systems that are supported by RCP and HIL software of dSPACE Release 2017-B. Refer to <a href="#">Operating System</a> on page 268.

<sup>1)</sup> The **Processor Interface** sublibrary of the RTI FPGA Programming Blockset also supports MATLAB R2017b.

### Enhancements to the DS1202 FPGA I/O Type 1 framework

The framework for MicroLabBox provides the following enhancement:

- The **Resolver** function lets you access a resolver sensor. With its help you can get the current angular position of a rotor.

For modeling the resolver access, refer to [Parameters Page \(FPGA\\_IO\\_READ\\_BL\)](#) ([RTI FPGA Programming Blockset - FPGA Interface Reference](#)).

For handcoding the resolver access, refer to [Resolver](#) ([RTI FPGA Programming Blockset - FPGA Handcode Interface Reference](#)).

### Enhancements to the frameworks of the DS2655 FPGA Base Board

FPGA applications that are modeled with function blocks for the DS2655 FPGA Base Board and for the I/O modules provide the following enhancements.

**Tuning of FPGA constants** You can adjust values of tunable FPGA constants with your experiment software. This feature must be enabled and specified before you build the FPGA application. Refer to [Adjusting Values of FPGA Constants](#) ([RTI FPGA Programming Blockset Guide](#)).

**Support of FPGA test access** FPGA applications support an FPGA test access. The FPGA test access lets you set values for the I/O interface and the processor interface with your experiment software. Refer to [How to Enable FPGA Test Access](#) ([RTI FPGA Programming Blockset Guide](#)).

**Support of new DS2655 FPGA Base Board variant** The *DS2655 (7K410) FPGA Base Board* framework now supports the DS2655 (7K410) FPGA Base Board.

All features of the *DS2655 (7K160) FPGA Base Board* framework are also supported by the new framework for the new board variant.

### General enhancements

**Improved processor interface blocks** The processor interface blocks of the Processor Interface library now support SCALEXIO systems. Refer to [How to Generate a Processor Interface](#) ([RTI FPGA Programming Blockset Guide](#)).

**UART demo project for SCALEXIO systems** A UART demo project for SCALEXIO systems provides a model block that simplifies the implementation of UART interfaces. Refer to [Modeling UART Communication](#) ([RTI FPGA Programming Blockset Guide](#)).

**Enhanced script interface** The script interface is enhanced with new script functions that you can use to automate the modeling and parametrizing of the FPGA applications. Refer to [Using Script Functions of the RTI FPGA Programming Blockset](#) ([RTI FPGA Programming Blockset Guide](#)).

### Related topics

#### Basics

[Migrating to RTI FPGA Programming Blockset 3.4](#) ..... 165

## Migrating to RTI FPGA Programming Blockset 3.4

### Introduction

There are various ways to migrate an existing model, depending on the blockset version used.



### Migrating from RTI FPGA Programming Blockset 1.1 and higher to 3.4

If you implemented your FPGA application with RTI FPGA Programming Blockset Version 1.1 and later and want to use it with RTI FPGA Programming Blockset 3.4, the framework automatically updates itself to the current framework version.

The update handles all the subsystems in the model/subsystem. The parameters of the blocks stay the same after updating to the current framework version.

#### Appearance of migrated processor interfaces with Goto and From blocks

With RTI FPGA Programming Blockset 3.1 ... 3.3 you modeled the processor interface of a SCALEXIO system with Simulink **Goto** and **From** blocks. If you migrate a model with **Goto** and **From** blocks, the update process migrates these blocks to the processor interface blocks of the Processor Interface library. The migration process does not change the size of the origin blocks to beware the block arrangement of your model. Therefore, the appearance of the migrated blocks is different to the default appearance of processor interface blocks. The following illustrations gives you an example.

Appearance after the migration	Default appearance
	

### ConfigurationDesk custom functions incompatible with dSPACE Release 2017-B

#### Note

Relevant for SCALEXIO systems with a DS2655 FPGA Base Board and a DS2655M1 Multi-I/O Module

An FPGA custom function block generated with RTI FPGA Programming Blockset 2.5 from dSPACE Release 2013-A and the real-time applications (\*.rta) containing the FPGA custom function block are incompatible with the current dSPACE Release. To produce a usable custom function, you have to rebuild the FPGA model by using the current RTI FPGA Blockset.

**Using different dSPACE hardware**

Using an FPGA model on different dSPACE hardware requires some model modifications. Refer to [Migrating to Another dSPACE Hardware](#) ( RTI FPGA Programming Blockset Guide).

# RTI LIN MultiMessage Blockset

## Where to go from here

## Information in this section

New Features of the RTI LIN MultiMessage Blockset 2.9 .....	167
Migrating to RTI LIN MultiMessage Blockset 2.9 .....	168

## New Features of the RTI LIN MultiMessage Blockset 2.9

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### New supported platform

The RTI LIN MultiMessage Blockset supports SCALEXIO systems with a DS6351 LIN Board. The DS6351 LIN Board provides eight LIN channels.

### Support of AUTOSAR System Template 4.3.0

The RTI LIN MultiMessage Blockset supports the AUTOSAR System Template based on AUTOSAR Release 4.3.0 for describing LIN networks.

Refer to [General Settings Page \(RTILINMM MainSetup\)](#) ([RTI LIN MultiMessage Blockset Reference](#)).

## Migrating to RTI LIN MultiMessage Blockset 2.9

---

### Changed MEX compiler support

The MEX compiler support of the RTI LIN MultiMessage Blockset has changed. In earlier blockset versions, the Microsoft Windows SDK compiler was required to build MEX functions. With RTI LIN MultiMessage Blockset 2.9, only the following compilers are supported:

- MinGW (GNU Compiler Collection (GCC 4.9.2)): In combination with MATLAB Releases 2016a, 2016b and 2017a.
- MinGW (GNU Compiler Collection (GCC 5.3.0)): In combination with MATLAB Release 2017b.
- Microsoft Visual Studio 2015 Pro: In combination with MATLAB Release 2016a, 2016b, 2017a and 2017b.

---

### Working with models from earlier RTI LIN MultiMessage Blockset versions

To reuse a model created with an earlier RTI LIN MultiMessage Blockset version, you must update the S-functions for all the RTILINMM blocks and save the model before modifying the LIN configuration.

To create new S-functions for all the RTILINMM blocks in your model in one step, you can perform one of the following actions after opening the model:

- In the MATLAB Command Window, enter `rtimmsu_update('System', gcs)`.  
For more information on the command and its options, enter `help rtimmsu_update` in the MATLAB Command Window.
- Select the **Create S-Function for all LIN Blocks** command from the **Options** menu of the RTILINMM GeneralSetup block.

For more information, refer to [Limitations of RTI LIN MultiMessage Blockset](#) ([RTI LIN MultiMessage Blockset Reference](#)).



# SCALEXIO Firmware

## New Features of the SCALEXIO Firmware 4.1

---

### **New supported hardware**

The SCALEXIO firmware supports the following new hardware:

- **SCALEXIO Processing Unit**  
A new version of a SCALEXIO Real-Time PC is supported that has an Intel® Xeon® processor E3-1275 v6 running at 3.8 GHz
- **DS2531 SCALEXIO SSD**  
A solid-state drive for storing data on a SCALEXIO system.
- **DS6202 Digital I/O Board**  
A standard SCALEXIO I/O board that provides 32 bidirectional digital I/O channels for signal measurement or signal generation.
- **DS6311 FlexRay Board**  
A standard SCALEXIO I/O board that provides 4 FlexRay channels for connecting a SCALEXIO system to FlexRay buses.
- **DS6341 CAN Board**  
A standard SCALEXIO I/O board that provides 4 CAN/CAN FD channels for connecting a SCALEXIO system to CAN or CAN FD buses.
- **DS6351 LIN Board**  
A standard SCALEXIO I/O board that provides 8 LIN channels for connecting a SCALEXIO system to LIN buses.
- **DS6551 IOCNET Link Board**  
A board that provides an additional external IOCNET downlink port in a SCALEXIO LabBox. As an extension of the DS6001 Processor Board, this port can be used for multiprocessor connections or for connecting additional I/O units.



# SYNECT

---

**Where to go from here****Information in this section**

New Features of SYNECT 2.4 .....	172
Migrating to SYNECT 2.4 .....	183

## New Features of SYNECT 2.4

### Where to go from here

### Information in this section

<a href="#">New General Features of SYNECT</a> .....	172
Provides an overview of SYNECT's new general features.	
<a href="#">New Features of Test Management</a> .....	176
The following improvements have been made for managing tests.	
<a href="#">New Features of Model and Signal &amp; Parameter Management</a> .....	178
The following improvements have been made for model, signal & parameter management.	
<a href="#">New Features of Workflow Management</a> .....	181
The following improvements have been made for workflow management.	

## New General Features of SYNECT

### Introduction

SYNECT provides the following new general features.

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

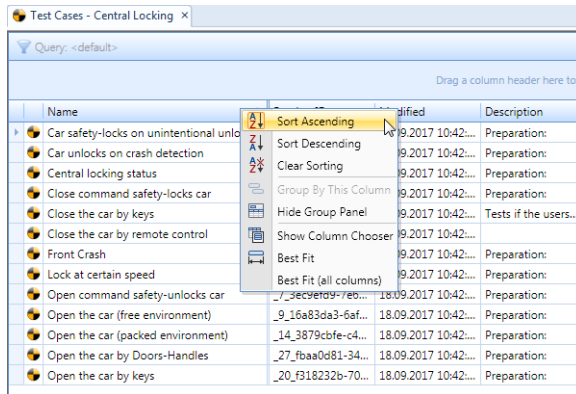
### Improved data grids

Data grids for items have been improved.

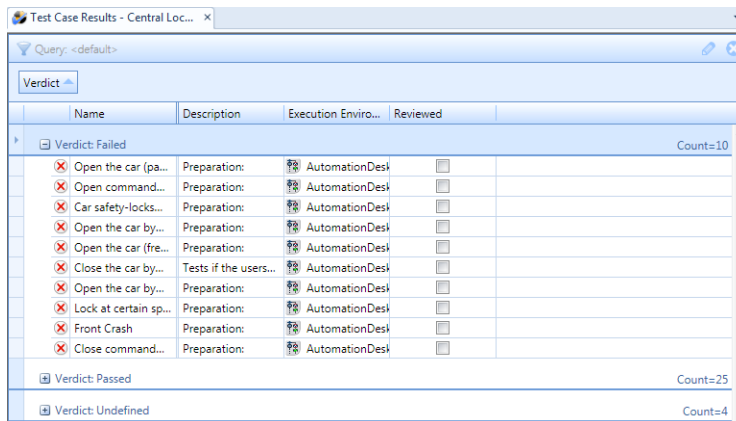
SYNECT now uses queries to search the information in the database, which is displayed in data grids. This improves the performance of data grids when large numbers of items are involved.

The handling of items in data grids with respect to sorting, grouping, and filtering has been adapted to the use of queries.

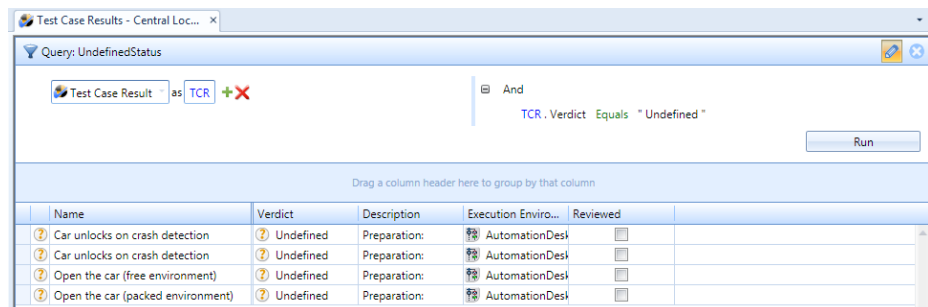
**Sorting** You can sort items in data grids according to the values of their attributes. SYNECT uses queries to get the attribute values from the database and sort them. Refer to the following illustration.



**Grouping** You can group items in a data grid according to the values of their attributes. SYNECT uses queries to get the attribute values from the database and group them. Refer to the following illustration.




**Filtering** You can use queries to filter for items in data grids. This lets you filter for items according to attribute values. You can also use more detailed queries and filter for items according to attributes of referenced items. SYNECT lets you load and save queries for each item type. Refer to the following illustration.



The following data grids have been improved:

- Test Cases
- Execution Plans

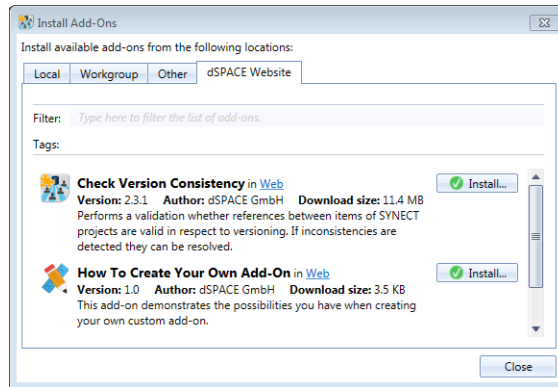
- **Add Items dialog (Execution Plan Details)**
- **Finished Executions**
- **Pending Executions**
- **Test Case Results** (all test case results of a project and test case results of a finished execution)

**Further reading** For details, refer to [Working with SYNECT](#) ( [SYNECT Guide](#)).

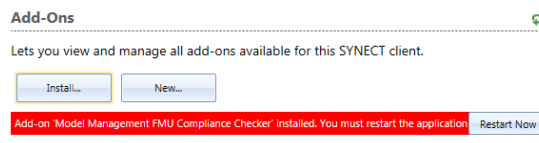
## Improved management of add-ons


Managing add-ons has been improved in the following aspects:

- The list of add-ons in the **Install Add-Ons** dialog and the list of installed add-ons has been improved in the following points:
  - The location of add-ons is displayed. This informs you where the add-on is located before the installation or, if installed, where it was installed.
  - You can filter the lists.
  - You can use tags to select add-ons. If you create an add-on, you can create tags to describe the add-on.
- You can now install add-ons that are provided on the dSPACE website. This lets you update SYNECT extensions with the newest add-ons. The download size of the add-ons is displayed in the **Install Add-Ons** dialog. Refer to the following illustration.



- You can now restart SYNECT after you installed an add-on.



**Further reading** For more information, refer to [Managing Add-Ons](#) ( [SYNECT Guide](#)).

## Release configurations

You can now create and specify release configurations.

This lets you specify reference types that are release-relevant, i.e., you can specify which additional items are released if you release an item.

**Default behavior** By default, all reference types are release-relevant. However, you can change this behavior and specify which reference types are release-relevant.

Link types can also be release-relevant. You can specify which link types are release-relevant when you create the link types. You can change this setting in a release configuration.

The following table shows the default release configuration of selected signal & parameter management items and the related link types of SYNECT's demo data:

Item Type	Outgoing References	Release-Relevant
Variable	<ul style="list-style-type: none"> <li>▪ Typedef</li> <li>▪ Scaling</li> <li>▪ Unit</li> </ul>	<ul style="list-style-type: none"> <li>▪ ✓</li> <li>▪ ✓</li> <li>▪ ✓</li> </ul>
VAR2REQ (link type)	Signal & Parameter Requirement (target item)	✓
Typedef	Scaling	✓
Scaling	Unit	✓
Unit	–	–

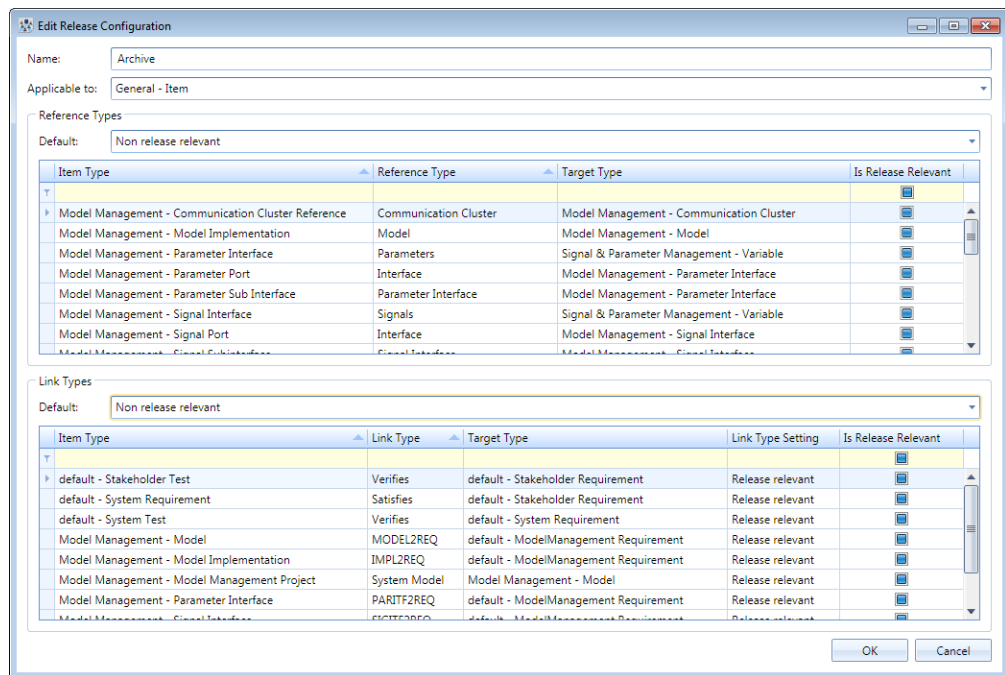
**Example release configuration** If you want to archive items, it can be useful to release selected items without releasing referenced items.

The following table shows an example release configuration that you can use to archive items.

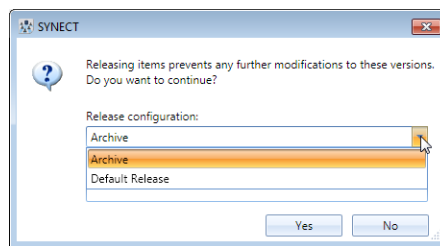
Item Type	Outgoing References	Release-Relevant
Variable	<ul style="list-style-type: none"> <li>▪ Typedef</li> <li>▪ Scaling</li> <li>▪ Unit</li> </ul>	<ul style="list-style-type: none"> <li>▪ –</li> <li>▪ –</li> <li>▪ –</li> </ul>
VAR2REQ (link type)	Signal & Parameter Requirement (target item)	–
Typedef	Scaling	–
Scaling	Unit	–
Unit	–	–

**Edit Release Configuration dialog** SYNECT provides the **Edit Release Configuration** dialog. It lets you specify release configurations. SYNECT's default behavior is preselected when you create a release configuration.

The following illustration shows an example release configuration that you can use to archive items.



**Releasing items** When you release an item, you can select a release configuration and control additionally released items.



**Further reading** For more information, refer to [Basics on Release Configurations](#) ([SYNECT Guide](#)).

## New Features of Test Management

### Introduction

The following improvements have been made for managing tests.

### Improved local executions queue

The local executions queue has been improved in the following points:

- If you select execution environments in the queue configuration, only executions with one of the selected execution environments are executed.



Executions with no execution environment or none of the selected execution environments are not considered for execution.

- If you select variant configurations in the queue configuration, only executions with one of the selected variant configurations are executed. Executions with no variant configuration or none of the selected variant configurations are not considered for execution.
- You can now configure the local executions queue to ignore the **Planned** attribute for execution by selecting the **Execute Future Executions** checkbox. If unchecked, SYNECT starts executions only when the planned date is reached or lies in the past. SYNECT waits for executions that are planned in the future and executes them when their planned date is reached. If selected, SYNECT starts executions in the order of their planned date, even if it has not been reached yet.

**Further reading** For more information, refer to [Executing Executions in a Queue](#) ([SYNECT Guide](#)).

## Improved data assignment

SYNECT's data assignment feature now supports binding variants separately for the test environment and the system under test.


When configuring the data pool to be used, you can specify which selected variant configuration maps the data pool.

SYNECT uses a data pool with the same name as the variant configuration selected for the planned execution.

The list of data pools that you can select in the **Data Assignment** dialog is limited by the following conditions:

- The data pools in the list must have the same name as variant configuration.
- The variant configuration that matches a data pool must comply with the variant configurations for the variant information, test environment, and test item of the planned execution. Constraints of the variant model must be met.

The following illustration shows how to configure the data pool to be used.

**Data Pool Usage** 

Lets you configure the settings for using the data pool. The 'Data pool to be used' option is evaluated during the 'Data Assignment' process before an execution starts. The mapping between a test case parameter and a data pool item can be configured via the 'Parameter assignment attribute' option.

**Data pool to be used:**

- Manual
- Selected Variant Information Variant Configuration
- Selected Test Environment Variant Configuration
- Selected Test Item Variant Configuration
- Custom Attribute

Configures the data pool to be used for preparing executions. If the Custom Attribute option is activated, a custom attribute of String type at the Execution item type has to be selected.

**Further reading** For more information, refer to [Parameterizing Tests](#) ([SYNECT Guide](#)).

## New Features of Model and Signal & Parameter Management

---

### Introduction

The following improvements have been made for model and signal & parameter management.

---

### Integrating system models

With this version, SYNECT provides a set of features that support system model integration.

System models, i.e., models of an ECU network and its environment, can be integrated from container files of the system model components. Additional items now represent communication clusters and the access of models to them. Import of container files is supported with plug-ins for V-ECU implementation container files, Simulink implementation container files, etc. Diagrams let you connect model components graphically. You can map signals of connected interfaces, if required, to specify valid port connections for model simulation.

Workflow management lets you perform complex tasks. For example, you can build system models for VEOS with a predefined workflow.

SYNECT focuses on the following scenarios for system model integration:

- Creating system models: To create a new system model from container files of the model components.
  - Updating system models: To update an existing model by importing updated container files of model components or adding new model components via import.
  - Building system models: To build a system model with VEOS for virtual validation.
- 

### SYNECT Add-on for System Model Integration

SYNECT provides an add-on for integrating system models.


The add-on simplifies the use cases that SYNECT supports for system model integration and virtual validation of models.

The add-on provides the following:

- A custom model item type for V-ECUs.
- A ribbon extension with additional commands that simplify the integration of system models.
- A workflow for building system models with VEOS.

The workflow exports the system model to a system model container file (SMC) that is imported to VEOS and used to build an offline simulation application (OSA). The classification of the OSA file is stored in SYNECT's database.

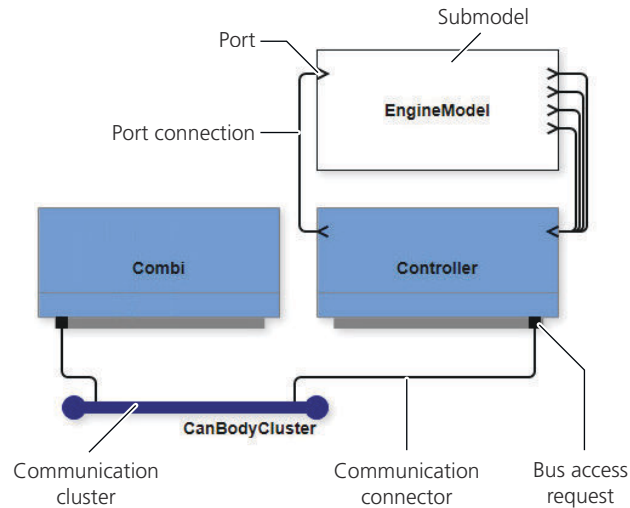
You can install the add-on via **Database – Extensions – Add-Ons**.

**Further reading** For more information, refer to [Integrating System Models](#) ( [SYNECT Guide](#)).

## Graphical modeling

You can now use diagrams for graphical modeling. You can create new model management items in the context of diagrams and add existing items to diagrams.

Diagrams let you connect models with other models via port connections.



**Further reading** For more information, refer to [Working with Diagrams](#) ([SYNECT Guide](#)).

## Specifying the data flow in composition models

To simulate a composition model, the data flow between the submodels must be specified, i.e., port connections must be valid.

Port connections are valid if one of the following conditions is fulfilled:

- Matching signal pairs of the involved port interfaces can be found on the basis of their names. This condition is fulfilled if you connect ports with the same interfaces.
- A signal mapping is used for the port connection. This type of mapping lets you define signal pairs.

SYNECT provides the **Signal Mapping** pane to map signals of port connections.

Signal Mapping			Signal Mapping		
Name	Typedef	Scaling	Name	Typedef	Scaling
PpRawSensors_Map.1	Float64 (1)		FuelsysSensors_RpRawSensors_Map	Float64 (1)	
			FuelsysSensors_RpRawSensors_Throttle	Float64 (1)	
			FuelsysSensors_RpRawSensors_Speed	Float64 (1)	
			FuelsysSensors_RpRawSensors_Ego	Float64 (1)	

**Further reading** For more information, refer to [Basics on Signal Mappings](#) ([SYNECT Guide](#)).

## Support for communication clusters

SYNECT provides items for communication clusters, communication connectors, and bus access requests. The items let you model the connection of a model to a communication cluster.

**Further reading** For more information, refer to [Connecting Models With Communication Clusters](#) ([📖 SYNECT Guide](#)).

### V-ECU implementation container support

You can now import V-ECU implementation containers to SYNECT. SYNECT creates related model and signal & parameter management items during the import.

SYNECT is compatible with the following V-ECU implementation container versions:

V-ECU Implementations Created With...	V-ECU Implementation Version
dSPACE Release 2017-B: <ul style="list-style-type: none"> <li>▪ SystemDesk 5.0</li> <li>▪ TargetLink 4.3</li> </ul>	2.6
dSPACE Release 2017-A: <ul style="list-style-type: none"> <li>▪ SystemDesk 4.8</li> </ul>	2.5
dSPACE Release 2016-B: <ul style="list-style-type: none"> <li>▪ SystemDesk 4.7</li> <li>▪ TargetLink 4.2</li> </ul>	2.4.1
dSPACE Release 2016-A: <ul style="list-style-type: none"> <li>▪ SystemDesk 4.6</li> </ul>	2.4

**Further reading** For more information, refer to [Basics on the V-ECU Implementation Container Plug-In](#) ([📖 SYNECT Guide](#)).

### Simulink implementation container support

You can now import Simulink implementation containers (SIC) to SYNECT. SYNECT creates related model, signal & parameter management items on import.

SYNECT is compatible with the following SIC versions:

SIC Files Created with Model Interface Package for Simulink of ...	SIC Version
dSPACE Release 2017-B (Model Interface Package for Simulink 3.5)	1.3
dSPACE Release 2017-A (Model Interface Package for Simulink 3.4)	1.2.1
dSPACE Release 2016-B (Model Interface Package for Simulink 3.3)	1.2
dSPACE Release 2016-A (Model Interface Package for Simulink 3.2)	1.1

**Further reading** For more information, refer to [Basics on the Simulink Implementation Container Plug-In](#) ([📖 SYNECT Guide](#)).

## New Features of Workflow Management

### Introduction

The following improvements have been made for workflow management.

### Workflow steps

- You can now define the how SYNECT reacts if the execution of a workflow step fails. This is useful for workflow steps that are optional for the overall success of a workflow.

You can select one of the following options:

Option	Description
<b>Ignore</b>	The subsequent steps of the workflow are executed. The workflow step result is ignored for the overall workflow result, i.e., if a step fails the overall workflow result can be <b>successful</b> .
<b>Continue</b>	The subsequent steps of the workflow are executed. If a workflow step fails, the overall workflow result is <b>failed</b> .
<b>Abort</b>	If a workflow step fails, the overall workflow is aborted. The overall workflow result is <b>failed</b> .

- You can now enter Python or MATLAB code in a workflow step. This is useful for simple or general-purpose code.
  - The code is stored in the database. SYNECT creates a temporary code file for step execution.
  - SYNECT provides a template that you can load into a workflow step. The template provides variables and a function frame.
  - You can import code from a file into a workflow step.

The screenshot shows the 'Workflow Steps' editor interface. On the left, a tree view shows a project named 'AR Fuelsys' with sub-items 'ExportSMC', 'BuildOSA', and 'ClassifyOSA'. The main editor area has tabs for 'General', 'Interface', 'Custom Attributes', and 'Code'. The 'Code' tab is active, displaying a Python function definition:

```

1 def CreateServerScriptArgs(WorkspaceId, ProjectId, ImplementationContextId, WorkingFolderPath, FilePath, Cla
2     """
3     Creates the arguments for the server script which classifies
4     the file in the working folder of the implementation.
5
6     @param WorkspaceId: Id of the workspace that contains the model management project.
7     @param ProjectId: Id of the model management project.
8     @param ImplementationContextId: Id of the model implementation context that shall be exported.
9     @param WorkingFolderPath: Full path to the working folder directory of the model implementation.
10    @param FilePath: Full path of the file to add classification information for.
11    @param Classification: The file classification to be added for the file.
12    """
13    import os
14    from os import path
15    FolderList = []
16    # get file path relative to working folder and split each folder name
17    if not path.dirname(FilePath) == WorkingFolderPath:
18        RelativeWorkingFolderPath = path.relpath(path.dirname(FilePath), WorkingFolderPath)
19        FolderList = RelativeWorkingFolderPath.split(os.sep)
20    # check if the file path is in the working folder
21    for FolderName in FolderList:
22        if FolderName == "..":
23            return None
  
```

At the bottom of the editor, there are buttons for 'Load Template', 'Import from File', and 'Store Changes', along with a 'Last modified' timestamp.

**Further reading** For more information, refer to [Workflow Steps](#) ([SYNECT Guide](#)).

**Jobs**

The **Variant Configuration Execution List** is now grouped by variant model subsets. This provides a better view on selected variant configurations for large variant models.

This applies to scheduled and immediate jobs.

**Scheduled jobs**

You can now repeat scheduled jobs for a time period.

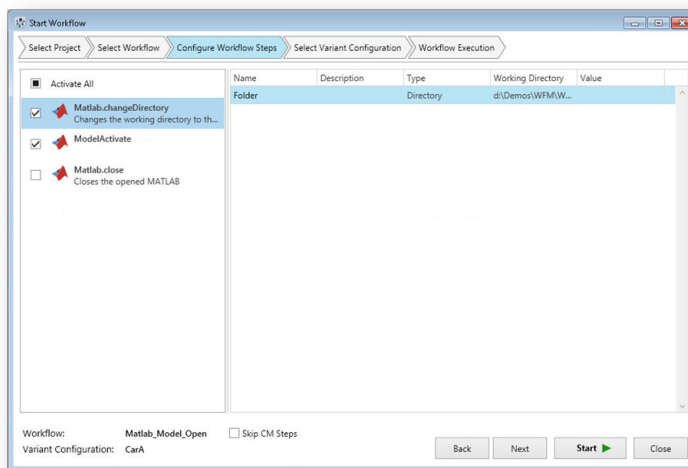
**Immediate jobs**

You can now use a filter to select workstations that execute an immediate job.

**Start Workflow wizard**

You can disable workflow steps in the **Start Workflow** wizard.

This is most useful when you test or debug a workflow. For example, you can disable a step that closes MATLAB if you plan to analyze an open model.

**Client API**

You can now use a settings object to specify parameters, such as the project ID and the workflow name, for executing a workflow or automating the **Start Workflow** wizard. Refer to [WFMWorkflowExecution / ISnWFMWorkflowExecution <<Interface>>](#) ([SYNECT Client API Reference](#)).

**Usability**

Copying parameters has been improved.

When you paste a copied parameter, you can now select from the following options:

- Copy and replace
- Do not copy
- Copy but keep both parameters

# Migrating to SYNECT 2.4

## Migrating Databases

---

### Introduction

To use the data from previous SYNECT versions with SYNECT 2.4, you have to migrate SYNECT's database.

To migrate databases of SYNECT 2.0 - 2.3 to SYNECT 2.4, SYNECT 2.4 provides the **Database Migrator**.

#### Note

Contact dSPACE Support if you want to migrate SYNECT versions prior to SYNECT 2.0.

For basic information and instructions on migrating databases, refer to [Migrating Databases from Previous SYNECT Versions](#) ([📖 The SYNECT Server Guide](#)).





# SystemDesk

---

**Where to go from here**

**Information in this section**

New Features of SystemDesk 5.0 .....	186
Migrating to SystemDesk 5.0 .....	193

# New Features of SystemDesk 5.0

## Where to go from here

## Information in this section

<a href="#">New General Features</a> .....	186
Provides information on new general features.	
<a href="#">Configuring ECUs</a> .....	188
Provides information on new features for configuring ECUs.	
<a href="#">Managing V-ECUs</a> .....	189
Provides information on improvements that were made for managing V-ECUs.	
<a href="#">Splittable AUTOSAR Elements</a> .....	190
Provides information on improvements that were made to support import scenarios that involve splittable AUTOSAR elements.	
<a href="#">Process Support With AUTOSAR Master Files</a> .....	192
Provides information on improvements that were made for process support with AUTOSAR master files.	

## New General Features

### AUTOSAR Releases supported by SystemDesk 5.0

AUTOSAR support has not been changed since SystemDesk 4.8. The following AUTOSAR Releases are supported:

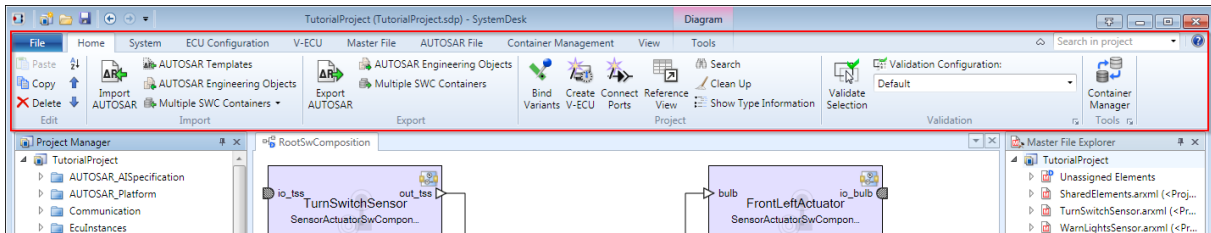
**Modeling support** SystemDesk supports the modeling of software and system architectures according to the AUTOSAR 4.3.0 Release.

**Data exchange support** SystemDesk supports AUTOSAR 4.3.0, 4.2.2, 4.2.1, 4.1.3, 4.1.2, 4.1.1, 4.0.3, and 4.0.2 for data exchange.

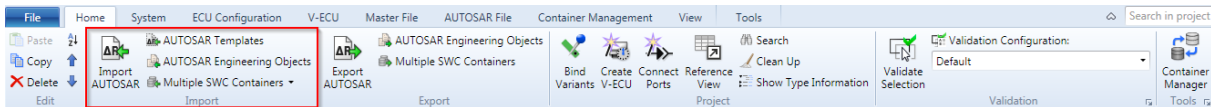
### More intuitive user interface

The user interface of SystemDesk is now more intuitive: Its menu bar and toolbar have been replaced by *ribbons*.

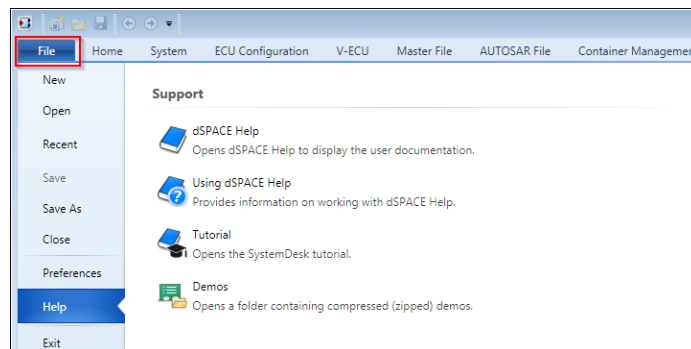
**Ribbons** SystemDesk's ribbons organize and group commands that belong together. They are located at the top of the user interface. Refer to the following illustration, which shows the **Home** ribbon:



Every ribbon features *ribbon groups*, each of which provides a set of related commands. Refer to the following illustration, which shows the **Import** ribbon group:



**Backstage view** SystemDesk's **File** ribbon or *Backstage view* provides basic commands, for example, for opening and saving SDP files. It also provides quick access to the recently used SDP files, SystemDesk's preferences, and the user documentation. Refer to the following illustration, which shows the **Help** ribbon group in the backstage view.



**Improved validation of AUTOSAR elements**

SystemDesk's validation feature has been improved in the following points:

- You can now assign validation rule configurations to specific tasks. This allows you to validate according to specific rule configuration when you perform tasks such as generating the RTE, exporting SWC containers, and exporting V-ECU implementation containers. SystemDesk provides standard tasks that you can select. You can also define your own tasks.

Task	Associated Configuration
Export SWC Container	TargetLink 4.2 Compatibility Check
Configure Com	dSPACE Com Stack
Generate RTE	dSPACE RTE
Export V-ECU Implementation Container	Default

- You can now specify rules that must be applied in a validation executed with SystemDesk's automation API.

This lets you apply specific validation rules even if they are not part of a selected rule configuration or not enabled.

- You can now customize the severity of validation rules for a rule configuration. The following illustration shows an example

Enabled	ID	Name	Description	Default Severity	Custom Severity	Element Types
<input checked="" type="checkbox"/>	R101	<b>Name Collision</b>	constr_2508: Two or more elements in a namespace have the same short name.	Error		IBaseElement
<input checked="" type="checkbox"/>	R102	<b>Autosar References</b>	Referenced elements have the wrong type or are unavailable.	Error		IBaseElement
<input type="checkbox"/>	R103	<b>Autosar Multiplicities</b>	Checks the multiplicity of elements according to the AUTOSAR STRICT schema.	Error		IBaseElement
<input checked="" type="checkbox"/>	R104	<b>Autosar Engineering Object File Subsistence</b>	Checks file references of AUTOSAR engineering objects. Only implementations with references to software components of application software and implementations without any software component reference are considered.	Warning	Error	IAutosarEngineeringObject

**Further reading** For more details, refer to [Validating SystemDesk Elements](#) (📖 SystemDesk Manual).

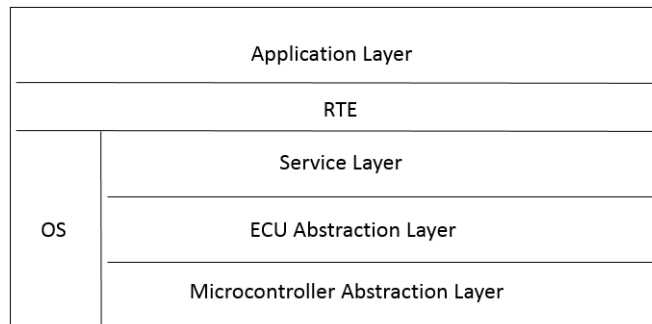
## Configuring ECUs

### Improved support for basic software modules for virtual validation

With this version, SystemDesk provides additional basic software modules of the microcontroller abstraction layer (MCAL).

This enables you to integrate, e.g., third-party diagnostic basic software modules of the service and ECU abstraction layer in V-ECUs by using SystemDesk's ECU configuration framework.

The following illustration shows the layered software architecture according to AUTOSAR.



**Support for dSPACE basic software modules** SystemDesk provides selected basic software modules for virtual validation.

You can perform the following actions for dSPACE basic software modules:

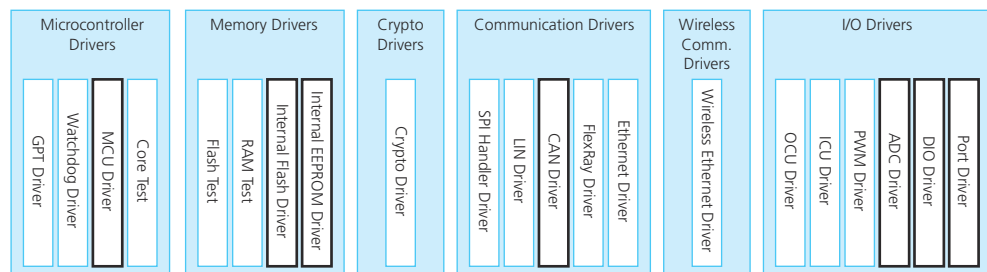
- Use upstream mappings to derive the module's configuration parameters from the system description as defined by AUTOSAR.
- (If required) Generate SWC descriptions.
- (If required) Generate code for simulation.

**Additional dSPACE basic software modules** The following additional basic software components are provided with this version of SystemDesk:

- Microcontroller drivers
  - MCU driver (Mcu module)
- Memory drivers
  - Internal Flash driver (Fls module)
  - Internal EEPROM driver (Eep module)
- Communication drivers
  - CAN driver (Can module)
- I/O drivers
  - ADC driver (Adc module)

The following illustration shows the microcontroller abstraction layer (MCAL) according to AUTOSAR. dSPACE provides basic software modules for the highlighted modules.

Microcontroller Abstraction Layer



#### Further reading

Refer to [Configuring ECUs](#) (📖 SystemDesk Manual).

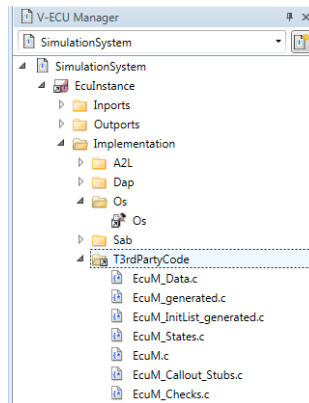
## Managing V-ECUs

#### Improved support for managing V-ECUs

The following improvements were made for managing V-ECUs:

- You can now add external code files to model-based V-ECU implementations. The external code files are included in the build of a related V-ECU and are also taken into account when you export a V-ECU implementation container.

You can select the files to add from the file system or from an ASAM 3.0 catalog file (CTLG).



- You can now specify that a code file of a V-ECU is included in another code file of the V-ECU and must not be compiled.

This lets you build V-ECUs from code files that make use of compiler directives, which include code files. If you do not use the new file category in these situations, the compiler also compiles the included file which can lead to an error due to duplicate symbols.

However, this applies only to code-based V-ECUs or model-based V-ECUs with external code files. V-ECU implementations that are generated by SystemDesk do not use the include file mechanism.

#### Further reading

Refer to [Creating Simulation Systems for Virtual Validation](#) ( SystemDesk Manual).

## Splittable AUTOSAR Elements

### Support for import scenarios that involve splittable AUTOSAR elements

With this version, SystemDesk supports import scenarios that involve splittable AUTOSAR elements.

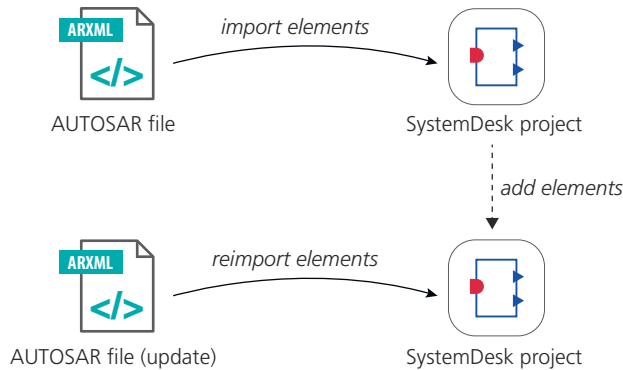
This lets you keep splittable AUTOSAR elements that you added in your working project when you reimport AUTOSAR files.

Suppose, you want to extend a system extract by implementing a software component in the following scenario:

1. A system extract is provided to you in one or more AUTOSAR files.  
You want to extend some of the contained AUTOSAR elements by adding new AUTOSAR elements that are splittable. Other AUTOSAR elements in the system extract are to remain unchanged.
2. You import the input AUTOSAR files to SystemDesk.
3. You work with SystemDesk and add AUTOSAR elements to your SystemDesk project.

4. At different points in time, updates of the input AUTOSAR files are provided to you. You reimport them to update the AUTOSAR elements in your working project.

The following illustration depicts the scenario:

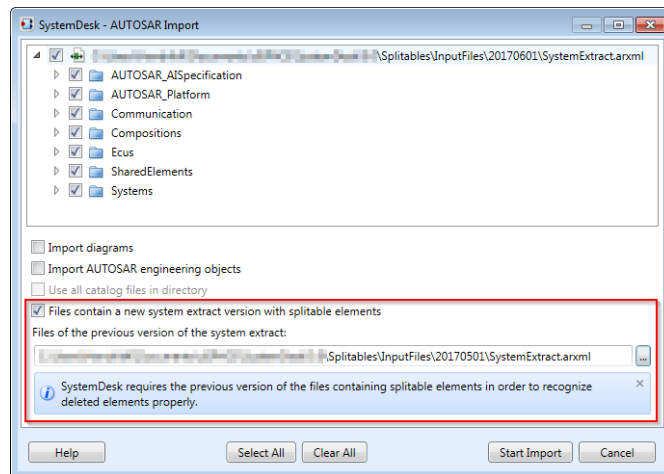


**Keeping splittable AR Elements** SystemDesk now lets you keep splittable elements that you added to a project instead of deleting them when the parent element is overwritten on import.

This applies to splittable elements such as the following:

- Port
- SWC internal behavior
- Runnable entity
- Per instance memory

The following illustration shows the **AUTOSAR Import** dialog that is configured to keep splittable elements during reimport.



### Further reading

For more details on working with SystemDesk's diagrams, refer to [Importing and Exporting AUTOSAR Files](#) (📖 SystemDesk Manual).

## Process Support With AUTOSAR Master Files

---

### Improved support for master files

The handling of AUTOSAR master files has been improved for the following points:

- You can load and save a master file list that contains the locations of master files.

This improves SystemDesk's support for multi-user scenarios. It lets you use a list of master files in different SystemDesk projects. This means, you can load a specific list of master files in different SystemDesk projects that work with the same AUTOSAR data. You can manage the list centrally and distribute it to the members of a working group.

- You can compare master files of the current SystemDesk session with the ones on your disk. This lets you identify differences between the AUTOSAR elements that you work with compared to the reference master files.


You can use an external compare tool with command line arguments from SystemDesk for comparing and viewing differences.

- You can specify the read-write behavior for master files that are read-only. You can specify to overwrite read-only files, skip read-only files, or to interact via a dialog.

This lets you protect selected master files from being overwritten accidentally.

---

### Further reading

For more information, refer to [Assigning AUTOSAR Elements to Master Files](#) ( [SystemDesk Manual](#)).



# Migrating to SystemDesk 5.0

## Migrating to SystemDesk 5.0

---

### Automatic migration

SystemDesk 5.0 automatically migrates SystemDesk 4.7, and 4.8 SDP project files during the loading process.

#### Note

You are recommended to install the most recent patch for SystemDesk 4.7 or 4.8. Then, save the SDP project files you want to migrate before opening them in SystemDesk 5.0.

---

### Migrating from SystemDesk 4.8

**Migrating scripts for automating SystemDesk** SystemDesk's API was changed with SystemDesk 5.0. Some interfaces were added with respect to SystemDesk 4.8. A number of interfaces were changed as well.

For information, refer to [API Changes from SystemDesk 4.8 to SystemDesk 5.0](#) ( [SystemDesk API Reference](#)).

---

### Migrating from SystemDesk 4.7

**Migrating scripts for automating SystemDesk** SystemDesk's API was changed with SystemDesk 4.8. Some interfaces were removed or added with respect to SystemDesk 4.7. A number of interfaces were changed as well.

For information, refer to [API Changes from SystemDesk 4.7 to SystemDesk 4.8](#) ( [SystemDesk API Reference](#)).



# TargetLink


---

## Where to go from here

## Information in this section

<a href="#">New Features of TargetLink 4.3 and TargetLink Data Dictionary 4.3.....</a>	196
<a href="#">Migrating to TargetLink 4.3 and TargetLink Data Dictionary 4.3 .....</a>	212
<a href="#">Changes in Future TargetLink Versions .....</a>	258

## Information in other sections

 [TargetLink New Features and Migration Guide](#)  
Provides information on new features, migration steps, discontinuations and code changes of the different TargetLink releases.

## New Features of TargetLink 4.3 and TargetLink Data Dictionary 4.3

### New licensing for dSPACE products

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### Where to go from here

### Information in this section

Modeling in Simulink or Stateflow .....	196
Code Generation Core Functionality .....	198
Modular Development .....	201
AUTOSAR .....	202
Target Simulation (PIL) .....	205
Data Dictionary and Data Management .....	206
Code Generator Options .....	206
API Functions and Hook Scripts .....	207
Other .....	207

## Modeling in Simulink or Stateflow

### Where to go from here

### Information in this section

Newly Supported Simulink Block .....	197
Bus-Capable Custom Code Block (Type II) .....	197
Other Simulink/Stateflow Features .....	197

## Newly Supported Simulink Block

---

**Support of the Simulink Delay block** TargetLink now supports the Simulink Delay block, which can be used to delay a signal by multiple fixed or variable sample steps.

**Related documentation** [Delay Block](#) ([📖 TargetLink Block and Object Reference](#))

## Bus-Capable Custom Code Block (Type II)

---

**Bus-Capable Custom Code Block (Type II)** TargetLink now supports the use of Simulink buses and TargetLink structured data types with the Custom Code (type II) block.

Related documentation:

- [Basics on Using Simulink Buses and TargetLink Structured Data Types with the Custom Code \(Type II\) Block](#) ([📖 TargetLink Preparation and Simulation Guide](#))
- [Overview of Methods for Preparing Unsupported Simulink Blocks for TargetLink Code Generation](#) ([📖 TargetLink Preparation and Simulation Guide](#))
- [Custom Code Block Description](#) ([📖 TargetLink Block and Object Reference](#))
- [BUS\\_CC](#) ([📖 TargetLink Demo Models](#))

## Other Simulink/Stateflow Features

---

**Support for resettable subsystems**

TargetLink now supports resettable subsystems. You can either use a Simulink Resettable Subsystem or copy&paste the Reset Port block in a subsystem. The subsystem then becomes resettable. The reset trigger signal resets all state variables in the blocks of the subsystem.

**Related documentation**

- [Basics on Resettable Subsystems](#) ([📖 TargetLink Customization and Optimization Guide](#))
- [Code-Relevant Simulink Blocks](#) ([📖 TargetLink Block and Object Reference](#))

**Full support for States when enabling behavior of function-call-triggered subsystems**

TargetLink now fully supports the `held` and `reset` settings of the **States when enabling** property for the Trigger block when the function-call trigger type is

selected. It lets you decide the way states are dealt with in function-call-triggered subsystems.

Until now, only the `inherit` setting was supported.

---

### Support of NXOR at Logical Operator block

TargetLink now supports the NXOR operation at the **Logical Operator** block.

---

### Support of Min/Max at bit operation blocks

The Code Generator now supports Min/Max constraints at the input and output signals of the following blocks:

- **Bit Clear**
- **Bit Set**
- **Bitwise Operator**
- **Extract Bits**
- **Shift Arithmetic**

---

### Improved TL\_Blackbox block mask type

TargetLink now supports masked subsystems with the **TL\_BlackBox** mask type to be placed anywhere in a model. TL\_BlackBox subsystems no longer have to reside in subsystems specified as external functions. In addition, the new `tLIIsCodeGenerationInProgress` API function lets you model TL\_BlackBox subsystems.

Related documentation:

- [Overview of Methods for Preparing Unsupported Simulink Blocks for TargetLink Code Generation](#) ( [TargetLink Preparation and Simulation Guide](#))

## Code Generation Core Functionality

---

### Where to go from here

### Information in this section

<a href="#">Code Decorations (Declaration Statements and Section Names) .....</a>	<a href="#">199</a>
<a href="#">MISRA C Compliance .....</a>	<a href="#">199</a>
<a href="#">Improved Support of Variable Vector Widths .....</a>	<a href="#">200</a>

## Code Decorations (Declaration Statements and Section Names)

### Code Decorations

TargetLink's mechanism of generating declaration statements and section names (i.e., code decorations) was improved.

You can now specify these decorations via dedicated objects in the Data Dictionary. This lets you filter the code elements to decorate by their characteristics (data type, width, initialization, scope, and storage).

Via the **AvoidStaticLocalScope** Code Generator option you can avoid variables of local scope with static storage duration.

#### Related documentation

- [Migrating Data Dictionaries to CodeDecorationSets](#) on page 219
- [Decorating Production Code](#) ([📖 TargetLink Customization and Optimization Guide](#))
- [CodeDecoration](#) ([📖 TargetLink Data Dictionary Reference](#))
- [AvoidStaticLocalScope](#) ([📖 TargetLink Block and Object Reference](#))

## MISRA C Compliance

### Rule compliance

It is now possible to generate code that complies with all rules classified as **mandatory** or **required** according to autocoding classification.

Related documentation

- [TargetLink's MISRA C:2012 Compliance Documentation document](#)

### Enable/disable pointer arithmetic in look-up table functions

TargetLink lets you enable/disable pointer arithmetic when generating code from Prelookup and Interpolation Using Prelookup blocks. If you set the **ImplementLookUpTableFunctionsWithPointerArithmetics** Code Generator option to **off**, pointer arithmetic is deactivated according to MISRA C:2012 rules. Instead, index accesses are used.

Keep in mind that this is not possible for 1D and 2D Look-Up Table blocks, because they can also be mapped with the Prelookup and Interpolation using Prelookup blocks.

#### Related documentation

- [Prelookup Block Description](#) ([📖 TargetLink Block and Object Reference](#))
- [Interpolation Using Prelookup Block Description](#) ([📖 TargetLink Block and Object Reference](#))
- [ImplementLookUpTableFunctionsWithPointerArithmetics](#) ([📖 TargetLink Block and Object Reference](#))

---

### Improvements to TargetLink's fixed-point library

Several improvements were made to TargetLink's Fixed-Point Library (DsFXP) to improve the compliance with individual MISRA C rules. The improvements include the following:

- TargetLink now supports 64-bit divisions without violating mandatory or required MISRA-C:2012 rules.
- **Implementation-defined behavior** messages reduced.
- The remaining **Implementation-defined behavior** messages are now documented.
- Less unreachable code (improvements to MISRA C:2012 - 2.1, 2.2, 14.3 rules).
- No more superfluous determination of loop variables (MISRA C:2012 18.1 rule).

#### Related documentation

- None
- 

### Explicit casts and floating point auxiliary variables for look-up table functions

To improve MISRA C compliance, the following changes were made to the code of look-up table functions in scenarios mixing fixed-point and floating point code:

- Implicit type conversions were made explicit via casts
- 

### Related topics

#### References

[ImplementLookUpTableFunctionsWithPointerArithmetics](#) (📖 TargetLink Block and Object Reference)

---

## Improved Support of Variable Vector Widths

---

### Supported blocks

TargetLink now supports variable vector widths for the **Assignment**, **Custom Code**, **Delay**, and **Selector** blocks. Specific settings need to be set.

#### Related documentation

- [Details on Variable Vector Width Implementation](#) (📖 TargetLink Customization and Optimization Guide)
- 

### Width macros referenced in the model

Width macros specified as DD variable objects can now be referenced in TargetLink blocks. This allows you to make the active width value of a width-variented DD variable explicitly available in the model.

#### Related documentation

- [Details on Variable Vector Width Implementation](#) (📖 TargetLink Customization and Optimization Guide)



- [Example of a Width Macro Referenced in the Model](#) (📖 TargetLink Customization and Optimization Guide)

### Report on terminated width propagation

By setting a specific code generator option, TargetLink generates a report on issues related to the propagation of variable vector widths throughout the model. The terminating blocks and ports are described.

#### Related documentation

- [How to Generate a Report on the Propagation Failures](#) (📖 TargetLink Customization and Optimization Guide)
- [ReportVariableVectorWidthDetails](#) (📖 TargetLink Block and Object Reference)

### Related topics

#### References

[ReportVariableVectorWidthDetails](#) (📖 TargetLink Block and Object Reference)

## Modular Development

### Where to go from here

#### Information in this section

[Improved A2L File Generation](#) ..... 201

[Improved Workflow for Distributed Development](#) ..... 202

## Improved A2L File Generation

### Incremental A2L File Generation

TargetLink can now incrementally generate A2L files. These files contain only the variable descriptions defined by the [Code generation unit \(CGU\)](#) (📖 Glossary), which is used to generate production code.

You can also generate an A2L file for several CGUs in one step. This file then contains the descriptions of the variables that are defined in the production code of any of the CGUs.

**Related documentation**    [Exchanging A2L Files](#) (📖 TargetLink Interoperation and Exchange Guide)

## Improved Workflow for Distributed Development

### Improved workflow

TargetLink now supports an improved workflow for distributed development.

This lets you perform the following actions:

- Use the same workflow with incremental code generation and model referencing
- Specify the location of the artifacts generated by TargetLink as required by your project
- Integrate existing artifacts into your project

### Related documentation

- [Specifying the Location of Artifacts Generated or Used by TargetLink](#) ([📖 TargetLink Customization and Optimization Guide](#))
- [Principles of Partitioning Models and Code](#) ([📖 TargetLink Customization and Optimization Guide](#))

## AUTOSAR

### Where to go from here

### Information in this section

<a href="#">Supported AUTOSAR Releases</a> .....	202
<a href="#">Memory Mapping</a> .....	203
<a href="#">Static Memories and Constant Memories for Measurement and Calibration</a> .....	203
<a href="#">Support for Rte_IsUpdated</a> .....	204
<a href="#">AUTOSAR Import and Export Improvements</a> .....	204

## Supported AUTOSAR Releases

### Supported AUTOSAR Releases

The following AUTOSAR Releases are supported:

AUTOSAR Release	Revision
4.3	4.3.0 <sup>1)</sup>
4.2	4.2.2 4.2.1

AUTOSAR Release	Revision
4.1	4.1.3
	4.1.2
	4.1.1
4.0	4.0.3
	4.0.2
3.2	3.2.3
	3.2.2
	3.2.1
3.1	3.1.5
	3.1.4
	3.1.2
	3.1.0
3.0	3.0.7
	3.0.6
	3.0.4
	3.0.2
2.1	2.1.4

<sup>1)</sup> New in TargetLink 4.3

## Memory Mapping

### SwAddressMethods and MemorySections

You can use `SwAddressMethods` and `MemorySections` for memory mapping as defined by AUTOSAR.

#### Related documentation

- [Mapping Variables and Functions to Memory Sections](#) ( TargetLink AUTOSAR Modeling Guide)
- [Decorating Production Code](#) ( TargetLink Customization and Optimization Guide)
- [AR\\_MEMORY\\_MAPPING](#) ( TargetLink Demo Models)

## Static Memories and Constant Memories for Measurement and Calibration

### Static memories and constant memories

TargetLink now supports static memories and constant memories for measurement and calibration as defined by AUTOSAR 4.x.

**Related documentation:**

- Basics on Static and Constant Memories in AUTOSAR 4.x ([📖 TargetLink AUTOSAR Modeling Guide](#))
- How to Model Static Memories for Measurement ([📖 TargetLink AUTOSAR Modeling Guide](#))
- How to Model Constant Memories for Calibration ([📖 TargetLink AUTOSAR Modeling Guide](#))

## Support for Rte\_IsUpdated

---

**Rte\_IsUpdated**

TargetLink now supports the **Rte\_IsUpdated** RTE API function.

**Related documentation**

- Basics on Checking the Update Flag of Data Elements in Explicit Sender-Receiver Communication ([📖 TargetLink AUTOSAR Modeling Guide](#))
- How to Model Checks of the Update Flag in Explicit Sender-Receiver Communication ([📖 TargetLink AUTOSAR Modeling Guide](#))

## AUTOSAR Import and Export Improvements

---

**Arrays of structs**

TargetLink can import and export AUTOSAR data prototypes that reference array of struct data types. The following AUTOSAR data prototypes are supported:

- Data elements (AUTOSAR 3.x/4.x)
- Interrunnable variables (AUTOSAR 4.x)
- Static memories (AUTOSAR 4.x)

Direct support for Array of Structs is available only for import and export. They are not supported in the model (as arrays of buses) and in code generation. For an example of how to use the imported data prototypes for modeling and code generation, see the **AR\_ARRAY\_OF\_STRUCT\_DATA** demo model.

**Related documentation**

- Array of Struct Support for Import and Export ([📖 TargetLink AUTOSAR Modeling Guide](#))
- AR\_ARRAY\_OF\_STRUCT\_DATA ([📖 TargetLink Demo Models](#))

**Related topics****Basics**

[AR\\_ARRAY\\_OF\\_STRUCT\\_DATA](#) ([📖 TargetLink Demo Models](#))

## Target Simulation (PIL)

### Changes in the Target Simulation Modules

#### New and discontinued compiler versions

The following table shows the compiler versions that are now supported by TargetLink 4.3, refer to the **New** and **No changes** columns. Compiler versions that are no longer supported are listed in the **Discontinued** column.

Microcontroller Family	Compiler	New	No Changes	Discontinued
ARM CortexM3	Keil	—	5.2	—
C16x	TASKING	—	8.6	—
MPC57xxVLE	Diab	—	5.9	—
	GreenHill	2016	—	2014
MPC560xVLE	Diab	—	5.9	—
	GreenHill	2016	—	2012,2014
RH850	GreenHill	2016	—	2015
S12X	Cosmic	—	4.8	—
	Metrowerk	—	5.1	—
SH2	Renesas	—	9.3	—
SH2A-FPU	Renesas	—	9.4	—
TriCore17xx	TASKING	6.2	3.2	6.0
TriCore2xx	TASKING	6.2		6.0
	GCC		4.6	
V850	GreenHill	2016		2015
XC22xx	TASKING		3.0	

For more information on the evaluation boards supported by TargetLink, refer to [Combinations of Evaluation Boards, Microcontrollers, and Compilers](#) ([📖 Evaluation Board Hardware Reference](#)).

#### Note

For more PIL support combinations that are part of a valid Software Maintenance Service (SMS) contract, refer to dSPACE's [TargetLink PIL Support](#) website at the [TargetLink Product Support Center](#).

## Data Dictionary and Data Management

### Further Improvements to the Data Dictionary

---

#### Empty Class property of a DD Variable object always causes an error in DD validation checks

The default `dsdd_validate` API command and the corresponding **Validate** function of the Data Dictionary Manager now always detect DD Variable objects with an empty `Class` property and return an error. In earlier versions of TargetLink, this might have been detected only during code generation.

### Code Generator Options

#### New Code Generator Options

---

##### Overview of new Code Generator options

The following new Code Generator options are available with TargetLink 4.3.

- **AvoidStaticLocalScope**

Specifies to avoid static storage duration for function-local variables.

- **ImplementLookUpTableFunctionsWithPointerArithmetics**


Enables the generation of pointer arithmetic in Look-Up Table functions, otherwise an index access is used where supported.

- **OutputMATLABCodeInfo**

Outputs XML files that contain information about MATLAB subfunctions and internal variables.

- **ReportVariableVectorWidthDetails**

When code is generated, a report on variable vector widths is generated. This report describes the blocks and ports where the propagation of **ExchangeableWidth** objects terminates and explains the reasons.

**Related documentation** For reference information on all Code Generator options, refer to [Alphabetical List of Code Generator Options](#) ( TargetLink Block and Object Reference).

##### Migration aspects of Code Generator options

Migration aspects include:

- Removed Code Generator option
- Changed Code Generator options
- Recommended compatibility settings
- Basics on changed defaults

For details, refer to [Migration Aspects Regarding Code Generator Options](#) on page 222.

## API Functions and Hook Scripts

### New API Functions

#### List of new API functions

API Function	Purpose
<code>tlCodeGenerationMetadata</code>	Saves and loads code generation meta data (DD <b>Subsystem</b> object).
<code>tlGetArtifactLocation</code>	Returns the location of the code generation unit's generated <a href="#">artifacts</a> ( <a href="#">Glossary</a> ).
<code>tlIsCodeGenerationInProgress</code>	Determines if the code generation process is active.
<code>tlPropman</code>	Command-line interface to the revised TargetLink Property Manager.

#### Related documentation:

- [tlPropman](#) ([TargetLink API Reference](#))
- [tlGetArtifactLocation](#) ([TargetLink API Reference](#))
- [tlCodeGenerationMetadata](#) ([TargetLink API Reference](#))
- [tlIsCodeGenerationInProgress](#) ([TargetLink API Reference](#))

## Other

#### Where to go from here

#### Information in this section

<a href="#">New Property Manager</a> .....	208
<a href="#">General Enhancements and Changes</a> .....	209
<a href="#">TargetLink Demos</a> .....	210

## New Property Manager

### New version of the Property Manger

TargetLink 4.3 provides the revised Property Manger to support you in exploring and changing the TargetLink properties of large models.

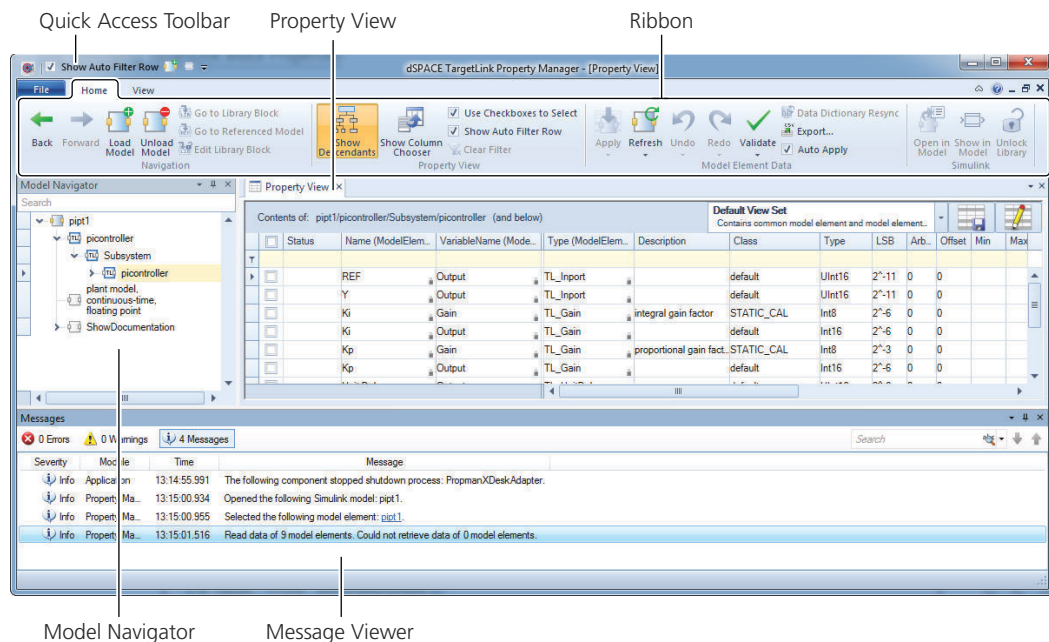
The Property Manager provides views for navigating through the Simulink model and for displaying and editing TargetLink properties of model elements.

Model elements can be, for example:

- Simulink subsystems
- TargetLink blocks
- Stateflow/MATLAB function objects

The Property Manager allows you to view and filter the properties of model elements and lets you simultaneously modify the same TargetLink property of several model elements. For Stateflow objects, this includes charts, events, states, and data.

The following illustration shows the new user interface of the Property Manager.



These are the main new user interface elements:



- The **Model Navigator** displays the hierarchy of models.

It lets you perform actions such as navigating through the model hierarchy and opening TargetLink dialogs of model elements (e.g. Stateflow objects). You can select a model element to display the TargetLink properties of this model element and (if applicable) underlying model elements in the **Property View**.



- The **Property View** is the working area in the Property Manager for displaying, editing, and validating the TargetLink properties of model elements and their variables. Each model element variable is displayed in an own row. You can perform actions such as focusing on specific properties and multi-editing property values.
- The **Message Viewer** displays system messages in a chronological order. It lets you search for messages and filters the messages to be displayed.
- The **Ribbon** organizes, groups, and labels the commands of the Property Manager.
- The **Quick Access Toolbar** is an easy way to call commands. You can customize it to contain the commands you use most frequently.

#### Related Documentation

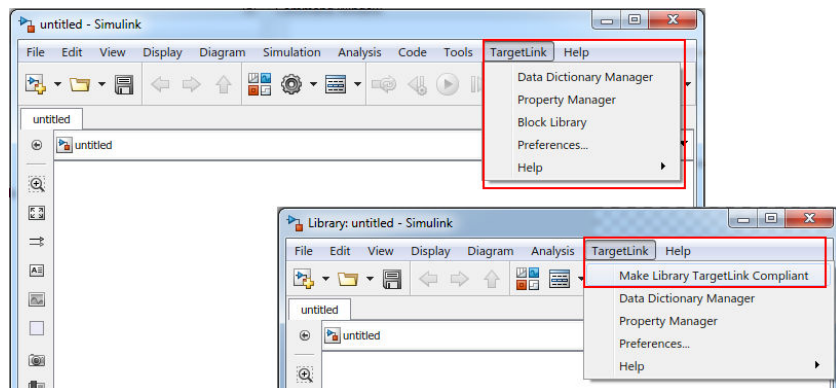
- [Modifying Multiple Properties at Once via the Property Manager](#) ( TargetLink Preparation and Simulation Guide)
- [Property Manager](#) ( TargetLink Tool and Utility Reference)
- [Property Manager](#) on page 226

## General Enhancements and Changes

### Improved user interfaces

The following improvements have been made to TargetLink's frontend:

**TargetLink menu is always visible** TargetLink now provides the **TargetLink** menu (with access to help, demos, etc.) in models and libraries that do not have TargetLink information (such as pure Simulink models). In addition, the **Make Library TargetLink Compliant** command is available for libraries.



**TargetLink Main Dialog tooltips** The TargetLink Main Dialog now provides more informative tooltips.

**Hyperlink for TargetLink File Export Utility** After a successful code generation, a new **Export** hyperlink in the MATLAB Command Window starts the **TargetLink File Export Utility**.

**Improvements to TargetLink block dialogs** TargetLink now provides access to Code Generator-relevant Simulink parameters in block dialogs of the following blocks:

- (Bus-) Inport
- (Bus-) Outport
- Data Store Read / Write
- Rate Limiter
- TargetLink Stateflow chart dialog

---

### Inactive MIL handler block

TargetLink lets you to comment out the MIL Handler block for simulations in order to simulate faster. When you start a simulation, TargetLink informs you that a MIL Handler block is commented out. Note that some TargetLink MIL simulation features like Overflow checking are only available with an active MIL Handler block.

---

### Support for A2L 1.6

TargetLink now supports Version 1.6.1 of the ASAM MCD-2 MC standard for A2L file export.

**Related documentation** [Exchanging A2L Files](#) ([📖 TargetLink Interoperation and Exchange Guide](#))

---

### Code Generation Progress dialog

TargetLink now displays a **Code Generation Progress** dialog when you start the code generation, including a **Cancel** button to abort a long running code generation.

**Related documentation** [Code Generation Progress Dialog](#) ([📖 TargetLink Tool and Utility Reference](#))

---

### New options in style definition file

TargetLink's style definition file has the following new options:

Option	Description
<code>&lt;TL:var-definition-comment show-default-scalings="false/true"&gt;</code>	Show scaling comments for variables with default scaling.
<code>&lt;TL:decldef-section-comment show="true/false"/&gt;</code>	Do not show information about variable class and bit width in the declaration comments.

## TargetLink Demos

---

### New demo models

The following new demo models are available for TargetLink 4.3:

**AR\_ARRAY\_OF\_STRUCT\_DATA** The demo model demonstrates exemplarily how to access array elements from an array of struct via RTE API calls based on a Custom Code (type II) block.

Refer to [AR\\_ARRAY\\_OF\\_STRUCT\\_DATA](#) ( [TargetLink Demo Models](#)).

**AR\_MEMORY\_MAPPING** The AR\_MEMORY\_MAPPING demo model demonstrates how to perform AUTOSAR memory mapping with TargetLink.

Refer to [AR\\_MEMORY\\_MAPPING](#) ( [TargetLink Demo Models](#)).

**BUS\_CC** The BUS\_CC demo model shows the use of Simulink buses and TargetLink structured data types with the Custom Code (type II) block.

Refer to [BUS\\_CC](#) ( [TargetLink Demo Models](#)).

**MODULAR\_DEVELOPMENT** The demo shows how the artifacts generated for incremental code generation units, such the production code or HTML documentation, can be integrated and used in an integration model.

Refer to [MODULAR\\_DEVELOPMENT](#) ( [TargetLink Demo Models](#)).

---

### Extended demo models

The following demo models have been extended for TargetLink 4.3:

**AR\_COLLISION\_DETECTION** The demo model has been extended to demonstrate the use of bus signals at the interface of a Custom Code (type II) block.

Refer to [AR\\_COLLISION\\_DETECTION](#) ( [TargetLink Demo Models](#)).

**CUSTOM\_ENHANCEMENT** The subsystem which is to be prepared now contains two Simulink **Tapped Delay** blocks.

Refer to [CUSTOM\\_ENHANCEMENT](#) ( [TargetLink Demo Models](#)).

# Migrating to TargetLink 4.3 and TargetLink Data Dictionary 4.3

---

**Upgrade process** Carefully read all of the following information and modify your tool chain accordingly, where necessary.

---

**Where to go from here**

**Information in this section**

MATLAB-Related Changes .....	212
General Migration Information .....	213
Migrating from TargetLink 4.2 to 4.3 .....	222
Code Changes .....	232
Discontinuations .....	255

## MATLAB-Related Changes

---

**Modified features since MATLAB R2016b**

**Changed startup behavior** MATLAB has changed the execution order of startup scripts.

- Up to MATLAB R2016a, the `startup.m` script is executed first, before the dSPACE-specific initialization scripts `dsstartup.m` and `dspoststartup.m`.
- As of MATLAB R2016b, the `startup.m` script is executed last, after the dSPACE-specific initialization scripts.

This modification might result in a changed startup behavior.

## General Migration Information

### Upgrade of Models, Libraries, and Data Dictionaries

#### Where to go from here

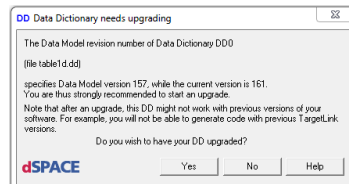
#### Information in this section

Basics on Migrating between TargetLink Versions .....	213
How to Upgrade a Data Dictionary with Included DD Files .....	216
How to Manually Upgrade Libraries and Models via the API .....	218
Migrating Data Dictionaries to CodeDecorationSets .....	219

### Basics on Migrating between TargetLink Versions

#### Automatic upgrade from TargetLink 3.1 or higher

TargetLink 4.3 automatically upgrades models, TargetLink-compliant libraries, and Data Dictionaries if they were created with TargetLink 3.1 or higher. You are prompted for the automatic upgrade when the Data Dictionary is opened with TargetLink 4.3 for the first time, for example:



The automatic upgrade comprises all the steps required by the single TargetLink versions in between. For example, an automatic upgrade from TargetLink 3.5 to TargetLink 4.3 comprises the steps 3.5 to 4.0 to 4.1 to 4.2 to 4.3.

#### Note

Check the TargetLink migration documentation of the different TargetLink versions whether user interaction is required.

**User interaction required** In the following cases, for example, the automatic upgrade requires additional user interaction:

- Libraries must be TargetLink-compliant. Otherwise, an upgrade is not possible at all.

- To upgrade DD files with included partial DD files, refer to [How to Upgrade a Data Dictionary with Included DD Files](#) on page 216.
- Style sheets for code generation are version-specific and subject to change from one TargetLink version to another. Thus, modified style sheets of older TargetLink versions have to be updated to match the current version (reapplying the modifications as intended).
- Custom code S-functions built with 32-bit TargetLink versions do not work with 64-bit versions of TargetLink and vice versa.  
Initiate a rebuild of all custom code S-functions using the `tlUpgrade('Model', <MyModel>, 'CheckModel', 'FixIssues')` API function.

### Making new libraries TargetLink-compliant

Libraries that you create from scratch, and that consist of TargetLink blocks, must be made upward-compatible so that you can upgrade them to a newer TargetLink version in the future. Otherwise, an upgrade will not be possible.

#### Note

A library does not become a TargetLink library just because it contains TargetLink blocks. The library itself must be TargetLink-compliant.

Refer to [How to Make TargetLink User Libraries Upgrade-Capable](#) ([📖 TargetLink Orientation and Overview Guide](#)).

### Making old libraries TargetLink-compliant

The following two approaches let you make libraries created with older TargetLink versions compliant with the current TargetLink version 4.3:

**The old TargetLink version is available** Use the old TargetLink version which the library was created with to make the library TargetLink-compliant. Refer to the TargetLink migration documentation of the old TargetLink version. You can then use this library with all the higher TargetLink versions because TargetLink automatically performs an upgrade. The library can still be used with TargetLink versions prior to TargetLink 4.3 because the automatic upgrade does not save a library in the newer TargetLink version.

**Only the current TargetLink version 4.3 is available** Use TargetLink version 4.3 and the `tlUpgrade` API command to make the library TargetLink-compliant. Refer to [How to Manually Upgrade Libraries and Models via the API](#) on page 218. Following this instruction the library is saved in TargetLink version 4.3 and, hence, cannot be used with TargetLink versions prior to TargetLink 4.3.

### Manual upgrade from TargetLink 2.x

Models, libraries, and Data Dictionaries created with TargetLink versions 2.x need to be upgraded manually to the highest TargetLink version 3.x (3.1...3.5) you have. Afterwards, an automatic upgrade is possible.

**Using TargetLink models with earlier TargetLink versions**

It is not possible to save models in the format of earlier TargetLink versions.

**Data model filter rule files**

Existing data model filter rule files can contain invalid elements, because the data model of the TargetLink Data Dictionary changed. The following files that were shipped with previous TargetLink versions can be affected:

- DD\_Filter\_Admin.xml
- DD\_Filter\_AR\_User.xml
- DD\_Filter\_NonAR\_NonRTOS\_User.xml

You can check filter rule files via the API in the MATLAB Command Window:

Checking a Single File	Checking Filter Rule Sets <sup>1)</sup>
<pre>dsdd_free; dsdd('ReadFilterRuleSet', 'file', '&lt;myFile&gt;.xml'); ds_error_register(dsdd('GetMessageList')); ds_msgdlg('update');</pre>	<pre>dsdd_free; dsdd('ReloadFilterRuleSets'); ds_error_register(dsdd('GetMessageList')); ds_msgdlg('update');</pre>

<sup>1)</sup> All the files contained in the directory defined in **Data Dictionary - Filter Rules** in the **TargetLink Preferences Editor**.

TargetLink informs you about errors in TargetLink's **Message Browser**. Each error contains the following information, so that you can fix it in an XML-capable editor of your choice:

- Filename
- Row number
- Column number

**Related topics**

**Basics**

[Basics on Code Formatting](#) (📖 TargetLink Customization and Optimization Guide)

**HowTos**

[How to Make TargetLink User Libraries Upgrade-Capable](#) (📖 TargetLink Orientation and Overview Guide)

[How to Manually Upgrade Libraries and Models via the API](#) ..... 218

**References**

[tlUpgrade](#) (📖 TargetLink API Reference)

## How to Upgrade a Data Dictionary with Included DD Files

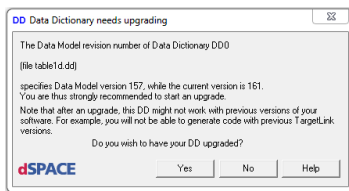
**Objective**

If you open a TargetLink model with an old Data Dictionary file that was not upgraded, you have to upgrade the Data Dictionary file.

**Method**

**To upgrade a Data Dictionary with included DD files**

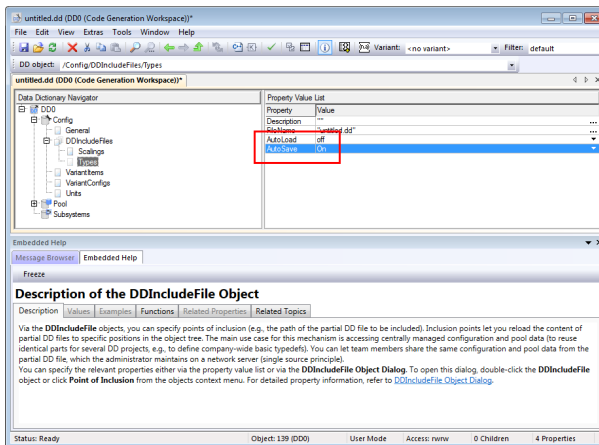
- 1 Open the model and the referenced TargetLink Data Dictionary, or type `dsdd('Open', <DDFile>)` in the MATLAB Command Window.  
The **Data Dictionary needs upgrading** dialog automatically opens if an earlier DD version is involved.



- 2 Select **No** in the upgrade dialog.



- 3 Under **/Config/DDIncludeFiles**, set the **AutoLoad** and **AutoSave** properties for each included DD file as shown in the following screenshot.

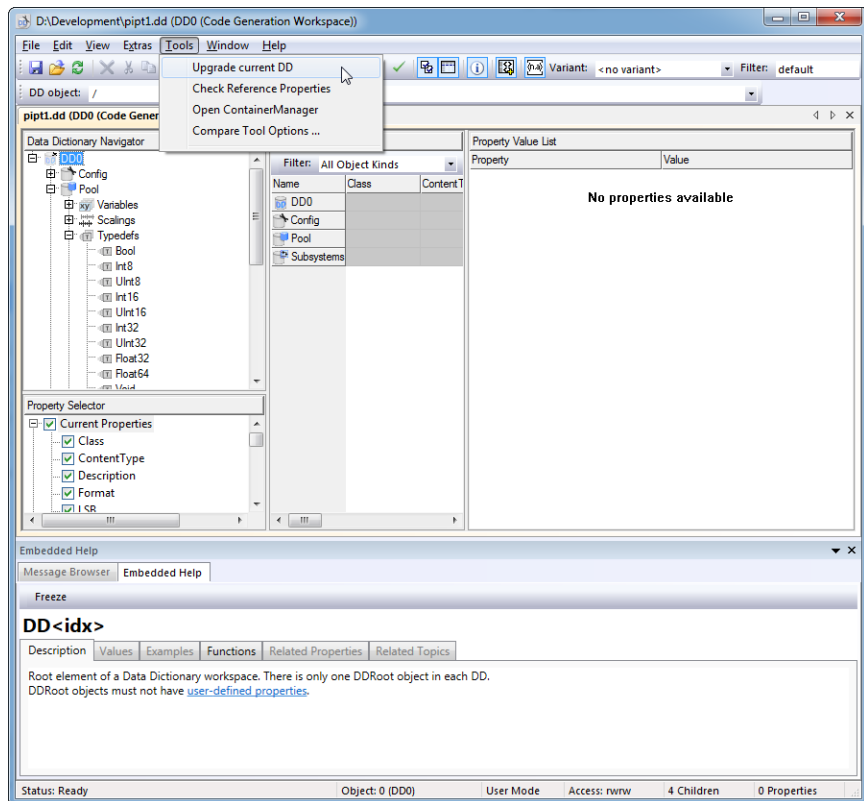


This ensures that after the Data Dictionary and the included DD files were upgraded, the included DD files that were upgraded are saved at the same time the Data Dictionary is saved. You can set these properties for a large number of included DD files via the **Object Explorer**.

#### Tip

You can also use the **Point of Inclusion** dialog to set the included DD file properties.

- 4 Start the Data Dictionary upgrade (with the included DD files) via **Tools – Upgrade current DD** in the DD Manager, or enter `dsdd('Upgrade')` in the MATLAB Command Window.



- 5 Save the Data Dictionary (with write permission to the relevant DD file). This completes the upgrade of the DD file and the included partial DD files.

## Result

When you open the DD file again, the upgrade dialog does not open, because the DD file and the included partial DD files are up-to-date. After the files were properly upgraded, you might want to restore the old settings for the included DD files.

## How to Manually Upgrade Libraries and Models via the API

### Objective

To prepare, for example, a central upgrade of libraries and models in a tool chain scenario with several users.

### Preconditions

The model or library files are available on MATLAB's search path but not opened.

The required and already upgraded DD project file is opened, for example, via `dsdd_manage_project('Open', '<name>.dd')`. Upgrade of DD project files is possible via `dsdd('Upgrade'[, <DD_Identifier>])`.

**Method**

**To manually upgrade libraries and models via the API**

1 Type this API command in the MATLAB Command Window:

```
tlUpgrade('Model', '<Model|Library>.mdl',
'CheckModel', 'FixIssues')
```

The model or library is upgraded.

**Note**

When upgrading models and libraries, first upgrade models or libraries that do not reference any other libraries, i.e., the blocks and subsystems they contain have no links to other libraries. Start with the bottom library and then upgrade the libraries above it in ascending order.

2 Save the upgraded model or library files, e.g., **Library.mdl**.

3 Repeat steps 1 and 2 for all other models or libraries.

**Result**

You upgraded your models and libraries.

**Related topics**

**References**

[tlUpgrade](#) (📖 TargetLink API Reference)

[Upgrade](#) (📖 TargetLink Data Dictionary Reference)

## Migrating Data Dictionaries to CodeDecorationSets

**Introduction of CodeDecorationSet and CodeDecoration objects**

TargetLink 4.3 introduces DD **CodeDecorationSet** and **CodeDecoration** objects.

Additionally, several properties were removed from the Data Dictionary's data model:

DD Object	Change	Replacement
FunctionClass	Removal of the <b>DeclarationStatements</b> and <b>SectionName</b> property.	The <b>DeclarationStatements</b> and <b>SectionName</b> property of the DD <b>CodeDecoration.Settings</b> object.
VariableClass		
VariableClassTemplate.Filter	Removal of the <b>WidthSpec</b> property.	The <b>WidthSpec</b> property of the DD <b>CodeDecoration.Filter</b> object.

**Automatic upgrade by TargetLink**

**Limitation** TargetLink no longer supports width-specific type prefixes for variable classes. The automatic upgrade of the Data Dictionary fails if your

original Data Dictionary contains variable class templates used to derive variable classes that have width-specific type prefixes.

Use declaration statements instead.

When opening a Data Dictionary whose data model is older than the latest revision, TargetLink prompts you to perform an automatic upgrade.

Object Kind	Trigger	Upgrade Action
VariableClass FunctionClass	DeclarationStatements or SectionName properties are set.	<ol style="list-style-type: none"> <li>1. Creating a DD <b>CodeDecorationSet</b> object.</li> <li>2. Creating a single <b>CodeDecoration</b> for each <b>CodeDecorationSet</b>. The settings of the <b>CodeDecoration</b> object and its child objects match the settings of the original objects.</li> <li>3. Referencing the <b>CodeDecorationSet</b> object at the original object.</li> </ol>
SubStructTemplate	Filter.VariableClass is set.	Transfer the values of the following properties from the variable class to the <b>SubStructTemplate</b> object's filter: <ul style="list-style-type: none"> <li>▪ <b>DeclarationStatements</b></li> <li>▪ <b>SectionName</b></li> <li>▪ <b>TypePrefix</b></li> </ul>
VariableClassTemplate	<ul style="list-style-type: none"> <li>▪ <b>Filter.FilterCondition</b> property is set to ALL_TRUE.</li> <li>▪ <b>Settings.VariableClass</b> references a DD <b>VariableClass</b> object whose <b>DeclarationStatements</b> or <b>SectionName</b> properties are set.</li> <li>▪ The <b>Filter.WidthSpec</b> property is set for this DD <b>VariableClassTemplate</b> object or for another <b>VariableClassTemplate</b> object whose <b>Filter.VariableClassSpec</b> property has the same value.</li> </ul>	<ol style="list-style-type: none"> <li>1. Create a new DD <b>VariableClass</b> object in /Pool/VariableClasses/Templates.</li> <li>2. Create a new DD <b>CodeDecorationSet</b> in /Pool/CodeDecorations/Templates.</li> <li>3. For each <b>VariableClassTemplate</b> object with the same value of the <b>Filter.VariableClassSpec</b> property, add a <b>CodeDecoration</b> object to the <b>CodeDecorationSet</b>.</li> <li>4. Specifying the <b>CodeDecoration</b> object as required.</li> <li>5. Referencing the <b>CodeDecorationSet</b> object at the <b>VariableClass</b> object created in step 1.</li> <li>6. Referencing the <b>VariableClass</b> object created in step 1 via the <b>VariableClassTemplate.Settings.VariableClass</b> property.</li> </ol>

### Special considerations for variable class templates

If you specified DD **VariableClassTemplate** objects whose **Filter.FilterCondition** property is set to ALWAYS or NEVER, TargetLink deletes the object's **Filter.WidthSpec** property during upgrade without compensation.

If you want to keep the property's value, set the DD **VariableClassTemplate** object's **Filter.FilterCondition** to ALL\_TRUE before upgrading the Data Dictionary.

**Limitation** TargetLink does not upgrade DD **VariableClassTemplate** objects whose **Filter.FilterCondition** property is set to ONE\_OR\_MORE or ALL\_FALSE.

## Cleaning up

The automatic upgrade retains the functionality that was specified in the old Data Dictionary. You can clean it up manually to reduce the number of objects in the new Data Dictionary.

**Merging width-specific variable classes** If your old Data Dictionary contained width-specific **VariableClassTemplate/VariableClass** objects, your new Data Dictionary still contains all these variable classes.

Because the width-specific information is now stored in DD **CodeDecoration** objects, you can manually reduce the number of **VariableClass** objects in your DataDictionary. For example, if you were using variable classes in the form of **<Name>\_<width>** you can replace them by a single variable class **<Name>** that references a suitable code decoration set.

Two methods are possible:

- Merging code decoration sets:
  1. Copy all the single DD **CodeDecoration** objects that were generated during the upgrade for each variable class called **<Name>\_<width>** into a single **CodeDecorationSet** object.
  2. Make each **CodeDecoration** object width-specific via its filter.
  3. Reference the resulting **CodeDecorationSet** object at the variable class **<Name>**.
- Using a code decoration set created for variable class templates:
  1. If your original **<Name>\_<width>** variable classes were referenced by variable class templates, the DD upgrade automatically created a width-specific code decoration set in **/Pool/CodeDecorationSets/Templates** for you to use.
  2. You can reference this code decoration set at the resulting variable class called **<Name>**.

### Note

Make sure to replace references from model elements to the variable classes called **<Name>\_<width>** with references to **<Name>**.

**Retarget variable class templates** After you merged the old width-specific variable classes, you can once again use them as the target of your variable class templates. You can then delete all the variable classes contained in **/Pool/VariableClasses/Templates** that were created during the upgrade.

**Simplifying user-specified scope reduction chains (SRC)** If you used a user-specified SRC to specify declaration statements or section names for variables with specific scopes, you can do the following:

1. Adjust the **Filter.ScopeSpec** property of the code decoration that belongs to the set referenced by the first variable class in your SRC (highest scope) as required.
2. Delete the other variable classes of your SRC.
3. If you also used your SRC to prevent static local variables, you can now use the **AvoidStaticLocalScope** Code Generator option instead.

**Remove obsolete variable class templates** Find DD `VariableClassTemplate` objects with the same value of the `Filter.VariableClassSpec` property and delete all but one.

**Changes in the generated production code**

Changes in `CodeDecorations` can influence the generated production code mainly in the following respects:

- Changed code comments
- Sorting of variable definitions

Refer to [Code Changes](#) on page 232.

## Migrating from TargetLink 4.2 to 4.3

**Where to go from here**

**Information in this section**

<a href="#">Code Generator Options</a> .....	222
<a href="#">API Functions and Hook Scripts</a> .....	225
<a href="#">AUTOSAR-Related Migration Aspects</a> .....	226
<a href="#">Other</a> .....	226

## Code Generator Options

### Migration Aspects Regarding Code Generator Options

**Removed Code Generator option**

The following Code Generator options were removed from TargetLink:

Removed Option
None

**Changed Code Generator options**

Code Generator Option	Old Default	New Default
None	-	-

**Recommended compatibility settings**      None

**Basics on changed defaults**

The settings of the Code Generator options are stored with the model (model-based option storage). In addition, you can store user-defined sets of Code Generator options in DD **CodegenOptionSet** objects (DD-based option storage). You can use DD **CodegenOptionSet** objects as a central source for overwriting and to replace the model-based option settings that had been used since TargetLink 4.1.

If a model-based option value equals the old default value, it is automatically changed to the new default value during the upgrade. If a DD-based option value equals the old default value, it is not changed to the new default value during the upgrade but keeps the old value.

**Option value = old default**    If Code Generator options equal default values in the former TargetLink version and the new TargetLink version uses modified default values, note the following points:

- Model-based option:  
If you want to keep the old default values, you must reset them manually.
- DD-based option:  
If you want to use the new default values, you must adjust them manually.

The following table is an example describing the impact of a TargetLink upgrade (TargetLink<sub>Old</sub> to TargetLink<sub>New</sub>) on three arbitrary option values: 9, 11, and 13. The table illustrates two basic migration scenarios:

- Scenario #1: New default = old default  
The default value of a Code Generator option has not changed in the new TargetLink version, i.e., the default value remains 9.  
None of the option values is changed.
- Scenario #2: New default ≠ old default  
The default value of a Code Generator option changed with the new TargetLink version, i.e., the default value changed to 11.

Option Storage	Option Value (TargetLink <sub>Old</sub> ) Default = 9	Option Value (≤ TargetLink <sub>New</sub> )	
		Default = 9 (Scenario #1)	Default = 11 (Scenario #2)
Model-based	9 <sup>1)</sup>	9 <sup>1)</sup>	11 <sup>2)</sup>
	11	11	11 <sup>1)</sup>
	13	13	13

Option Storage	Option Value (TargetLink <sub>Old</sub> ) Default = 9	Option Value ( $\leq$ TargetLink <sub>New</sub> )	
		Default = 9 (Scenario #1)	Default = 11 (Scenario #2)
DD-based	9	9	9 <sup>3)</sup>
	11	11	11
	13	13	13

<sup>1)</sup> The option value is not stored with the model because it equals the default.

<sup>2)</sup> Manual reset might be necessary.

<sup>3)</sup> Manual adjustment might be necessary.

**Option value = new default** If the Code Generator options did not equal default values in the former TargetLink version (A) but in the new TargetLink version (B), TargetLink assumes that you intentionally specified the default value in the new TargetLink version. The same applies if the default changes again in the next TargetLink version (C).

#### Note

Upgrading TargetLink<sub>A</sub>  $\Rightarrow$  TargetLink<sub>B</sub>  $\Rightarrow$  TargetLink<sub>C</sub> and upgrading TargetLink<sub>A</sub>  $\Rightarrow$  TargetLink<sub>C</sub> can cause different option values (see the following table).

If the default values for TargetLink versions A, B, and C read 9, 11, and 13, and an option value equaled 11 in version A, an upgrade to version C changes the option value as follows:

Upgrade Strategy	Option Value TargetLink <sub>A</sub> Default = 9	Option Value TargetLink <sub>B</sub> Default = 11	Option Value TargetLink <sub>C</sub> Default = 13
A $\Rightarrow$ B $\Rightarrow$ C	11 ( $\neq$ default)	11 (= default) <sup>1)</sup>	13 (= default) <sup>1)</sup>
A $\Rightarrow$ C	11 ( $\neq$ default)	—	11 ( $\neq$ default)

<sup>1)</sup> The option value is not stored with the model, because it equals the default.

#### New Code Generator options

For more details on new Code Generator options, refer to [New Code Generator Options](#) on page 206.



## API Functions and Hook Scripts

### Changes in TargetLink and TargetLink Data Dictionary API Functions

#### **t1SimInterface**

The **t1SimInterface** API function now provides two commands that let you use TargetLink's online parameter modification with AUTOSAR calibration parameters:

Command	Purpose
<b>DisableRteStartBySimulationSFcn</b>	Disables a call of the AUTOSAR <b>Rte_Start()</b> function when the simulation of the simulation application (SIL/PIL) starts. TargetLink automatically enables a call of the <b>Rte_Start()</b> function after simulation (SIL/PIL) finishes. The <b>Rte_Start()</b> function is called only if AUTOSAR code is simulated.
<b>EnableRteStartBySimulationSFcn</b>	Enables a call of the AUTOSAR <b>Rte_Start()</b> function when the simulation of the simulation application (SIL/PIL) starts. This applies only if the call of the AUTOSAR <b>Rte_Start()</b> function was previously disabled by the <b>DisableRteStartBySimulationSFcn</b> command.

#### **Related documentation**

- [Basics on Modifying Parameter Values for Simulation](#) ([📖 TargetLink Preparation and Simulation Guide](#))
- [How to Provide Manual Parameter Updates via the MATLAB API](#) ([📖 TargetLink Preparation and Simulation Guide](#))
- [t1SimInterface](#) ([📖 TargetLink API Reference](#))

#### **t1Upgrade**

**t1Upgrade** now lets you specify the TargetLink version to that you want to upgrade.

#### **Related documentation**

- [t1Upgrade](#) ([📖 TargetLink API Reference](#))

#### **Adapting the new workflow for distributed development**

**Migration issue** With TargetLink 4.3, the old workflow of distributed development for referenced models is no longer available. The related API functions **t1\_distribute\_refmodel\_files** and **t1\_integrate\_refmodel\_files** were removed from TargetLink.

**Solution** To distribute and integrate the files, specify their locations in the Data Dictionary.

To create separate development frame models, use **t1ExtractSubsystem**.

#### **Related documentation**

- [Specifying the Location of Artifacts Generated or Used by TargetLink](#) ([📖 TargetLink Customization and Optimization Guide](#))
- [t1ExtractSubsystem](#)

**Related topics**

**References**

[tSimInterface](#) (📖 [TargetLink API Reference](#))

## AUTOSAR-Related Migration Aspects

### Other

**Where to go from here**

**Information in this section**

<a href="#">Property Manager</a> .....	226
<a href="#">Various Migration Aspects</a> .....	228

## Property Manager

**Major changes**

See the following table for the main differences between the old version and the new version of the Property Manager.

Previous Property Manager	New Property Manager
<p><b>Block Filter</b> to specify the block types to be displayed.</p> <p>Separate <b>Property Editor</b> to edit property values.</p> <p><b>Property Selector</b> to define which properties of the block types selected in the Block Filter are to be displayed. Additionally, you can select and clear properties that are shown in the Block Explorer.</p>	<p>Different new filter options of the <b>Property View</b>. For more information, refer to <a href="#">How to Filter and Sort Property Values in the Property View</a> (📖 <a href="#">TargetLink Preparation and Simulation Guide</a>).</p> <p>Editing property values directly in the respective cell in the <b>Property View</b>. For more information, refer to <a href="#">Editing Property Values in the Property View</a> (📖 <a href="#">TargetLink Preparation and Simulation Guide</a>).</p> <p><b>Column Chooser</b> to select the model elements to be displayed in the <b>Property View</b>. For more information, refer to <a href="#">Column Chooser Dialog</a> (📖 <a href="#">TargetLink Tool and Utility Reference</a>).</p>

Previous Property Manager	New Property Manager
Displaying multiple models, each model is displayed in its own window.	Display multiple models and/or libraries simultaneously as individual nodes in the <b>Model Navigator</b> . For more information, refer to <a href="#">How to Load Models or Libraries into the Model Navigator</a> ( <a href="#">TargetLink Preparation and Simulation Guide</a> )
Status bar to display comments and information about the TargetLink/Simulink blocks, and how many TargetLink block types are selected in the Block Filter.	<b>Messages</b> control bar to display system messages in a chronological order. It provides a history of all the info, advice, error, and warning messages. This helps you check the system state. It lets you search for messages and filters the messages to be displayed. For more information, refer to <a href="#">Message Viewer</a> ( <a href="#">TargetLink Tool and Utility Reference</a> ).
TargetLink API names of properties as column headers.	Describing terms as column headers, because identical properties of different model elements are merged in one column. Hover over a cell in the <b>Property View</b> to display the property's TargetLink API name as a tooltip. The tooltips of header cells also provide a description of the property. For more information, refer to <a href="#">Display of Properties in the Property View</a> ( <a href="#">TargetLink Preparation and Simulation Guide</a> ). In the <b>Column Chooser</b> you can search for the property's TargetLink API name and a description is provided as a tooltip. For more information, refer to <a href="#">Column Chooser Dialog</a> ( <a href="#">TargetLink Tool and Utility Reference</a> ).
TargetLink internal content of TargetLink subsystems is hidden.	TargetLink internal content of TargetLink subsystems is displayed, for example, TargetLink simulation frames including additional S-functions. You cannot edit TargetLink internal data in the Property Manager.
Invalid inputs are rejected immediately.	Advanced validation of data. For more information, refer to <a href="#">Validation of Property Values</a> ( <a href="#">TargetLink Preparation and Simulation Guide</a> )

#### Related Documentation

- [New Property Manager](#) on page 208
- [Modifying Multiple Properties at Once via the Property Manager](#) ([TargetLink Preparation and Simulation Guide](#))
- [Property Manager](#) ([TargetLink Tool and Utility Reference](#))

#### Changes concerning the CSV Export

There are some structural changes in the CSV export of the new Property Manager:

- Only values containing special characters or line breaks are written in quotation marks.
- The first three columns always are:
  - Status
  - Name (ModelElement)
  - VariableName (ModelElement)



- Columns are renamed and columns with identical properties of different model elements are merged. For example, **gain.value** and **input.value** are now both called **InitialValue**.
- Line breaks in model element names are included in the CSV export.

### Dialogs hidden in the background

**Migration issue** If you open TargetLink dialogs in the Property Manager, the Simulink model window might not display in the foreground but in the background. Instead, the Windows taskbar displays a flashing message referring to the Simulink model window.

**Hint** Keep in mind that additional windows might open in the background unexpectedly. Arrange the window panes in a way that they are displayed side by side without covering each other.

#### Related documentation

- [Open Model](#) ( [TargetLink Tool and Utility Reference](#))
- [How to Trace Model Elements Back to the Simulink Model](#) ( [TargetLink Preparation and Simulation Guide](#))

### Reduced access rate of TargetLink API functions

**Migration issue** If a model is loaded in the Property Manager, the access rate of TargetLink API functions (for example, `tl_get` or `tl_set`) could be reduced. This can also affect the performance of tool chains.

**Solution** Close the Property Manager via the `tlPropman('Exit')` API command before the execution of tool chains.

For more information, refer to **tlPropman**.

### Related topics

#### References

[tlPropman](#) ( [TargetLink API Reference](#))

## Various Migration Aspects

### Recommended signal delay modelings with the newly supported Delay block

**Migration issue** The Delay block is supported as of TargetLink 4.3.

In new models, use the Delay block with appropriate settings rather than the following modeling options:

- Unit Delay Reset Enabled block
- Chain of Unit Delay blocks
- Custom Code block (type II) which has a Simulink Delay block under its mask

For the Unit Delay Reset Enabled block, take into account the following differences between these blocks:

Delay block	Unit Delay Reset Enabled block
<p>One output</p> <p>Reset behavior: The internal condition is <i>not</i> reset if all of the following conditions apply:</p> <ul style="list-style-type: none"> <li>▪ The enable signal is 0.</li> <li>▪ The <b>External reset</b> is not set to <b>None</b>.</li> <li>▪ The reset signal triggers a reset.</li> </ul> <p>If a state reset is successfully triggered, the initial value (IV) will be written to the output and the data signal (U) will be written to the state.</p>	<p>Two outputs (one for the delayed signal and one for the state signal)</p> <p>Reset behavior: The internal condition is reset if the following two conditions apply:</p> <ul style="list-style-type: none"> <li>▪ The enable signal is 0.</li> <li>▪ The reset signal should trigger a reset.</li> </ul> <p>If a state reset is successfully triggered, the initial value (IV) will be written to the output and the state.</p>

#### Related documentation

- [Delay Block](#) (📖 TargetLink Block and Object Reference)
- [CUSTOM\\_ENHANCEMENT](#) (📖 TargetLink Demo Models)

## Variable descriptions in A2L files

**Migration issue** A2L files generated with TargetLink 4.3 contain descriptions of only calibratable/displayable variables that are used in the production code of the code generation unit (CGU) for which the A2L file was generated.

**Solution** To include descriptions of variables that are used in different CGUs in a single A2L file, generate code for all the CGUs and generate one A2L file from all of the DD **Subsystem** objects.

To include descriptions of externally defined variables in the A2L file, select the **External variables** checkbox.

#### Related documentation

- [Basics on Specifying the Variables to Export to A2L Files](#) (📖 TargetLink Interoperation and Exchange Guide)
- [Export as A2L File](#) (📖 TargetLink Data Dictionary Manager Reference)

## Encoding of A2L files

**Migration issue** With Version 1.6.1 of the ASAM MCD-2 MC standard, A2L files now have to be encoded as UTF-8 with BOM.

To ensure downward compatibility, the A2L files generated by TargetLink are not encoded as in previous TargetLink versions according to the OS Locale (default) or character encoding setting specified in the DD.. This might cause problems with [measurement and calibration systems](#) (📖 Glossary) that require the A2L files to be encoded as specified by Version 1.6.1 of the ASAM MCD-2 MC standard.

**Solution** If your calibration and measurement system requires the encoding as specified by Version 1.6.1 of the ASAM MCD-2 MC standard, open the generated A2L file in an editor that supports conversion to UTF-8 with BOM

and save it accordingly (or use a command line tool that provides such conversion functionality).

#### Related documentation

- [Overview of the Supported Character Sets](#) ([📖 TargetLink Interoperation and Exchange Guide](#))
- [How to Specify the Used Character Set](#) ([📖 TargetLink Interoperation and Exchange Guide](#))

---

### Installing multiple TargetLink instances

By default, you can no longer install multiple instances of the same TargetLink version. This also applies to versions with different patch levels. You can still install different TargetLink versions on the same host PC.

If you need to install several instances of TargetLink 4.3, contact dSPACE Support.

---

### Changes in code output configuration

**Migration issue** With TargetLink 4.3, the code output configuration file (`cconfig.xml`) was changed.

**Reason** To clarify the comments of code statements.

**Solution** If you changed the code output configuration in a prior version, adapt your changes to the new version supplied with TargetLink 4.3.

**Related documentation** [Customizing TargetLink's Code Formatting](#) ([📖 TargetLink Customization and Optimization Guide](#))

---

### New build directory and location of RTE frame files

With TargetLink 4.3, the default for the build directory changed from `.\TlSim\<Application>\<Board>_<Compiler>` to `.\TLBuild\<Application>\<Board>_<Compiler>`.

The build folder is used to gather the files of the RTE frame, which are not CGU-specific. These files are used only for simulating the AUTOSAR compliant code.

---

### Changed behavior with scope reduction and initialization

**Migration issue** TargetLink's initialization behavior for variables changed. This applies to restart initialization and initialization at definition. If the variables are reduced along a user-specified scope reduction chain (SRC) that ends in a module-local variable class, TargetLink behaves as shown in the following table:

TargetLink ≤ 4.2	TargetLink 4.3
Removal of variable/restart initialization if a reduction to automatic storage duration is possible.	Variable/restart initialization even if a reduction to automatic storage duration is possible.

**Reason** TargetLink now treats user-defined SRCs that do not end with a variable class that has automatic storage duration as intended by the user: For example, to work with variable accesses by address, that are not visible to TargetLink.

**Solution** Resolve potential problems as follows:

No Reason for Shortened SRC	Reason for Shortened SRC but Initialization not Necessary
Change the module-local variable class as follows: <ul style="list-style-type: none"> <li>▪ <b>Optimization</b> = SCOPE_REDUCIBLE</li> <li>▪ <b>ScopeReducedClass</b> = &lt;&gt;<sup>1)</sup></li> </ul>	Change the variable classes forming the SRC as follows: <ul style="list-style-type: none"> <li>▪ <b>InitAtDefinition</b> = off</li> <li>▪ <b>RestartFunctionName</b> = ""</li> </ul>

<sup>1)</sup> Or references a variable class with static storage duration.

**Interpolation in look-up table blocks**

The **Look-Up Table** and **Look-Up Table (2D)** block support the **Use Input Nearest** look-up method again.

**Changed value of FilePath property**

**Migration issue** With TargetLink 4.3 the value of the **FilePath** property of DD **Module** objects contained in DD **Subsystem** objects changed. The value now contains the placeholder **\$(CGURootOutputFolder)** that is not evaluated by the **dsdd('GetFilePath',...)** DD API function.

**Reason** To make artifact locations user-specifiable.

**Solution** Use the **dsdd('GetCompiledFilePath',...)** DD API function instead.

**1x1 signals**

**Migration issue** During code generation, TargetLink displays the **E03004** message when a signal whose dimension is **1x1** is fed into one of the following blocks:

- **FIR Filter**
- **Discrete Filter**
- **Discrete Transfer Fcn**
- **Discrete-Time Integrator**

**Reason** Bug fix

**Solution** Place Simulink Reshape blocks on the signal lines connected to the TargetLink block and set their **Output dimensionality** to **1-D array**.

**Tip**

Simulink's Selector block outputs a **1x1** matrix signal when selecting a single element of a matrix or vector.

**TL\_ExternBlock mask type is no longer supported**

**Migration issue** With TargetLink 4.3, the **TL\_ExternBlock** mask type is no longer supported.

**Reason** The **TL\_BlackBox** mask type was improved.

**Solution** Use the **TL\_BlackBox** mask type instead. Adapt the mask type changes to the new version supplied with TargetLink 4.3.

**Related documentation** [Overview of Methods for Preparing Unsupported Simulink Blocks for TargetLink Code Generation](#) (📖 [TargetLink Preparation and Simulation Guide](#))

## Code Changes

### Where to go from here

### Information in this section

64-Bit Multiplication .....	232
AUTOSAR .....	234
Efficiency .....	236
Function Reuse .....	240
MISRA Compliance .....	241
Mixed Operations (Floating-Point and Fixed-Point Types) .....	242
State Reset .....	244
Other .....	250

## 64-Bit Multiplication

### Additional functions and macros for 64-bit multiplication

TargetLink's Fixed-Point Library now provides additional functions and macros for 64-bit multiplications that are used if one operand has a width smaller than 32-bit.

TargetLink ≤ 4.2	TargetLink 4.3
This can result in a different parameter order:	
<pre>F__I64MULU32U32((UInt32)UInt16Var,   UInt32Var, &amp;Aux_c_hi, &amp;Aux_c_lo);</pre>	<pre>F__I64MULU32U16(UInt32Var,   UInt16Var, &amp;Aux_c_hi, &amp;Aux_c_lo);</pre>



TargetLink ≤ 4.2	TargetLink 4.3
Operation are used as arguments: <sup>1)</sup>	
<pre>Aux_S32 =   ((Int32) Sa1_F32In) &lt;&lt; Ca1_ShiftVar; F_I64MULI32I32(Aux_S32,   (Int32) Ca1_I16Var, &amp;Aux_S32_a, &amp;Aux_U32);</pre>	<pre>F_I64MULI32I16(((Int32) Sa1_F32In) &lt;&lt;   Ca1_ShiftVar, Ca1_I16Var, &amp;Aux_S32, &amp;Aux_U32);</pre>
Constant literals can now have smaller fitting data types with casts:	
<pre>F_I64MULI32U32(Int32Var,   (UInt32) 250000000, &amp;Aux_hi, &amp;Aux_lo);</pre>	<pre>F_I64MULI32I32(Int32Var,   (Int32) 250000000, &amp;Aux_hi, &amp;Aux_lo);</pre>
The data type of the cast is determined by the constant's value. This can result in positive values becoming signed. Depending on the setting of the <b>DecimalConstants</b> Code Generator option, they might be generated in hexadecimal format. <sup>1)</sup>	
<pre>C_I64MULI32U32(Sa1_Gain[0],   (UInt32) 12582912, Aux_S32, Aux_U32);</pre>	<pre>C_I64MULI32I32(Sa1_Gain[0],   (Int32) 0xc00000, Aux_S32, Aux_U32);</pre>

<sup>1)</sup> This also applies to existing functions.

**Reasons**

- MISRA C compliance
- Code efficiency

**Migration issue**    None

**Accumulator of FIR filter**

The code for calculating the accumulator of FIR filters changed.

TargetLink ≤ 4.2	TargetLink 4.3
Calculation was protected, even without saturation, which sometimes resulted in an additional 64-bit auxiliary variable and a corresponding 64-bit multiplication macro.	The superfluous macros are no longer generated.
<pre>/* accumulation */ Aux_S32 = 0; Aux_S16 = *(pDelayLine++); Aux_S32_a = *(pCoeff++); F_I64MULI32I32((Int32) Aux_S16,   Aux_S32_a, &amp;Aux_S32_b, &amp;Aux_U32); Aux_S32 += ((Int32) Aux_U32); if (pDelayLine &gt; (X_Sa5_FIR_Filter + 4)) {   pDelayLine = X_Sa5_FIR_Filter; }</pre>	<pre>/* accumulation */ Aux_S32 = 0; Aux_S32 +=   (((Int32) *(pDelayLine++)) *   *(pCoeff++)); if (pDelayLine &gt; (X_Sa5_FIR_Filter + 4)) {   pDelayLine = X_Sa5_FIR_Filter; }</pre>

**Reason**

- MISRA C compliance
- Code efficiency

**Migration issue**    None

## AUTOSAR

### AUTOSAR compiler abstraction for multiple instantiation

When generating compiler abstraction macros for the instance handle parameter of runnables, the CONSTP2CONST macro now has **AUTOMATIC** as second parameter, not **RTE\_CONST**.

TargetLink ≤ 4.2	TargetLink 4.3
<pre>FUNC(void, Controller_CODE)  controller_runnable(CONSTP2CONST(Rte_CDS_Controller, RTE_CONST, RTE_CONST) instance)</pre>	<pre>FUNC(void, Controller_CODE)  controller_runnable(CONSTP2CONST(Rte_CDS_Controller, <b>AUTOMATIC</b>, RTE_CONST) instance)</pre>

**Reason** AUTOSAR compliance

**Migration issue** None

### AUTOSAR 4.x memory mapping

Preprocessor instructions for memory mapping in AUTOSAR 4.x code are now block statements.

TargetLink ≤ 4.2
<pre> /*****\   ARRunnable4: Default function class for AUTOSAR runnables \*****/ #define Controller_START_SEC_CODE #include "Controller_MemMap.h" FUNC(void, Controller_CODE) controller_runnable(Rte_ActivatingEvent_controller_runnable activation); #define Controller_STOP_SEC_CODE #include "Controller_MemMap.h"  #define Controller_START_SEC_CODE #include "Controller_MemMap.h" FUNC(void, Controller_CODE) linearization_runnable(void); #define Controller_STOP_SEC_CODE #include "Controller_MemMap.h"  #define Controller_START_SEC_CODE #include "Controller_MemMap.h" FUNC(void, Controller_CODE) Init_Controller(void); #define Controller_STOP_SEC_CODE #include "Controller_MemMap.h"</pre>

**TargetLink 4.3**

```
#define Controller_START_SEC_CODE
#include "Controller_MemMap.h"
/*****\
  ARRRunnable4: Default function class for AUTOSAR runnables
\*****/

FUNC(void, Controller_CODE) controller_runnable(Rte_ActivatingEvent_controller_runnable activation);

FUNC(void, Controller_CODE) linearization_runnable(void);

FUNC(void, Controller_CODE) Init_Controller(void);

#define Controller_STOP_SEC_CODE
#include "Controller_MemMap.h"
```

**Reason**

- Suppress superfluous code
- Readability

**Migration issue**    None

**Matrix parameters of runnables and operation calls**

Matrix actual parameters of runnable and operation call functions are now by default generated with non-global scope. This might result in the following changes:

- A parameter of the `Rte_Call` RTE API function, is no longer moved to a per instance memory.
- Different naming because `$R` is evaluated differently.
- Different naming if the local parameter is removed by optimization.
- Range propagation is now possible, which can result in simpler control flow and changes in saturation and arithmetic operations by optimization.

**Reason**    Code efficiency

**Migration issue**    None

**#include of Rte\_Type.h**

TargetLink's behavior for generating an `#include` of `Rte_Type.h` in header files that contain declarations of runnables has changed:

TargetLink ≤ 4.2	TargetLink 4.3
Always generated the include.	<p>Generates the include to <code>Rte_Type.h</code> only in the following cases:</p> <ul style="list-style-type: none"> <li>▪ A runnable's declaration uses macros for compiler abstraction, which are defined in <code>Compiler.h</code>. This header file in turn is included in <code>Rte_Type.h</code>.</li> <li>▪ A parameter uses one of the <code>Rte_Type.h</code> data types.</li> </ul> <p>If none of the conditions is fulfilled, TargetLink does not generate the include in the header file. In most cases, the include is generated in the corresponding source file, where it is needed if the runnable's implementation makes use of data types contained in <code>Rte_Type.h</code>, such as <code>Std_ReturnType</code>.</p>

**Reasons**

- Code efficiency
- Suppress superfluous code

**Migration issue** None

**NvData and reduced write operations**

The code generated for DD **NvDataElement** objects has changed, whose **ReduceWriteOperations** property is set to **on**. This results in better optimization of the **if** statement:

TargetLink ≤ 4.2	TargetLink 4.3
<pre> /* TL-Inport */ ptr = Rte_IRead_x(); &lt;SomeCode&gt; a = ... /* TL-Outport */ ptr = Rte_IRead_x(); if (ptr[1] != a) {     Rte_IWriteRef_x()[1] = a; }                     </pre>	<pre> /* TL-Inport */ ptr = Rte_IRead_x(); &lt;SomeCode&gt; a = ... /* TL-Outport */ if (ptr[1] != a) {     Rte_IWriteRef_x()[1] = a; }                     </pre>

**Reason** Code efficiency

**Migration issue** None

## Efficiency

**Restart functions generated for external subsystems**

Restart functions generated for external subsystems are now called at different intervals.

TargetLink ≤ 4.2	TargetLink 4.3
Called the restart function once per instance.	Calls the restart function once per CGU.

**Reason** Code efficiency

**Migration issue** None

**Improved code optimization for Discrete-Time Integrator block**

The code pattern for the block was optimized for some edge cases:

- It can now calculate in a smaller width
- Superflous saturation code can now be better detected and omitted in the production code

**Reason** Code efficiency

**Migration issue** None

**Propagation of initial values for matrix variables**

TargetLink's optimization now also propagates initialization values of matrix variables.

**TargetLink ≤ 4.2**

```
static Bool Sa1_DFFMat_NQ[3][5] =
{
  {
    /* [0][0..4] */ 1, 1, 1, 1, 1
    /* 1., 1., 1., 1., 1. */
  },
  {
    /* [1][0..4] */ 1, 1, 1, 1, 1
    /* 1., 1., 1., 1., 1. */
  },
  {
    /* [2][0..4] */ 1, 1, 1, 1, 1
    /* 1., 1., 1., 1., 1. */
  }
};

for (Aux_S32 = 0; Aux_S32 < 3; Aux_S32++)
{
  for (Aux_S32_a = 0; Aux_S32_a < 5; Aux_S32_a++)
  {
    /* Sink: Subsystem/Sink3 */
    Sa1_Sink3[Aux_S32][Aux_S32_a] = Sa1_DFFMat_NQ[Aux_S32][Aux_S32_a];
  }
}
```

**TargetLink 4.3**

```
for (Aux_S32 = 0; Aux_S32 < 3; Aux_S32++)
{
  for (Aux_S32_a = 0; Aux_S32_a < 5; Aux_S32_a++)
  {
    /* Sink: Subsystem/Sink3 */
    Sa1_Sink3[Aux_S32][Aux_S32_a] = 1;
  }
}
```

**Reason** Code efficiency

**Migration issue** None

### Slimmer code for expression +1/expression -1 and casts

TargetLink's optimization now performs several simplifications:

TargetLink ≤ 4.2	TargetLink 4.3
$(x + 1) - 1$	x
$(1 + x) - 1$	
$(x - 1) + 1$	
For integers, relations with 1 or - 1 are simplified as follows:	
$x - 1 < y$	x <= y
$x < y + 1$	
$x < 1 + y$	
$x <= y - 1$	x < y
$x + 1 <= y$	
$1 + x <= y$	
$x > y - 1$	x >= y
$x + 1 > y$	
$1 + x > y$	
$x - 1 >= y$	x > y
$x >= y + 1$	
$x >= 1 + y$	
For index expressions the topmost casts are omitted if not needed:	
$x[(\text{Int32}) (a + 1)]$	$x[a + 1]$

**Further simplification** TargetLink further simplifies expressions like  $z = x - 1$  that occur in a context that allows for any of the simplifications shown in the table above. Because TargetLink knows that the right side of the expression will be replaced by an access to x, TargetLink treats the cost of propagating the right side as a cost of an access to x.

TargetLink ≤ 4.2	TargetLink 4.3
<pre>z = x - 1; if (z &lt; y) {     m = z; } else {     m = y; }</pre>	<pre>if (x &lt;= y) {     m = x - 1; } else {     m = y; }</pre>

In the example above, the first occurrence of  $x - 1$  is simplified to  $x <= y$  in accordance with the first table. This lowers the cost of propagating the right side of  $z = x - 1$ ; into the if statement.

#### Reasons

- Code efficiency
- Readability

**Migration issue** None

**Reset when enabled flags for enabled subsystems**

TargetLink no longer generates an unnecessary *reset when enabled* flag (RSWE) if an **IF** block is placed in an enabled subsystem.

**TargetLink ≤ 4.2**

```

if (Sa1_EnableIn > 0) {
    /* call of function: TL_Root/Enable */
    Sa2_Enable(Sa1_InPort2, Sa1_InPort3, Sa1_InPort1);
    /* set system state to 'enabled' */
    Sa2_RSWE = 1;
}
else {
    if (Sa2_RSWE == 1) {
        /* set system state to 'disabled': TL_Root/Enable */
        Sa2_RSWE = 0;
        /* set block state to 'disabled': If: TL_Root/Enable/ResetIf */
        Sa2_ResetIf_LastSystem = 0;
        /* set block state to 'disabled': If: TL_Root/Enable/Else/Else/ResetIf */
        Sa5_ResetIf_LastSystem = 0;
    }
}

```

**TargetLink 4.3**

```

if (Sa1_EnableIn > 0) {
    /* call of function: TL_Root/Enable */
    Sa2_Enable(Sa1_InPort2, Sa1_InPort3, Sa1_InPort1);

}
else {
    /* set system state to 'disabled': TL_Root/Enable */
    Sa2_RSWE = 0;
    /* set block state to 'disabled': If: TL_Root/Enable/ResetIf */
    Sa2_ResetIf_LastSystem = 0;
    /* set block state to 'disabled': If: TL_Root/Enable/Else/Else/ResetIf */
    Sa5_ResetIf_LastSystem = 0;
}

```

**Reason** Code efficiency

**Migration issue** None

**tl\_defines.h**

TargetLink no longer generates the **TL\_FX\_GROUND** macro if it is not needed. This might result in **tl\_defines.h** no longer being generated.

**Reason** Suppress superfluous code

**Migration issue** None

**Multiplication with integer fractions (N/D)**

The code changed for multiplication with integer fractions ( $N/D = 2^Z$ ) now may sometimes be more precise with the same efficiency:

TargetLink ≤ 4.2	TargetLink 4.3
<pre>Sa1_OutPort =   (Int32) ((Sa1_InPort * 91) &gt;&gt; 15);</pre>	<pre>Sa1_OutPort =   (Int32) ((Sa1_InPort * 2913) &gt;&gt; 20);</pre>

**Reason** Precision  
**Migration issue** None

## Function Reuse

**Instance-specific variables and sub-reuse structs**

With TargetLink 4.3, instance-specific variables now might be placed in the same sub reuse struct. Accordingly, the number of sub reuse struct might change or the name of the sub reuse struct might appear to have changed. This change can result in mixed structures that contain volatile and nonvolatile variables.

The following code shows the definition of the structured data types of the reuse struct that results from a reused subsystem that contains two **Gain** blocks. One has its **Class** set to **default**, the other to **GLOBAL**:

TargetLink ≤ 4.2	TargetLink 4.3
<b>Example 1</b>	
<pre>typedef struct tagISV_Sx1_0_tp {   Int16 Sx1_Gain;   Int16 Sx2_Sum; } ISV_Sx1_0_tp;  typedef struct tagISV_Sx1_1_tp {   Int16 Sx1_Gain1; } ISV_Sx1_1_tp;  typedef struct tagISV_Sx1_tp {   ISV_Sx1_0_tp * pISV_Sx1_0_tp;   ISV_Sx1_1_tp * pISV_Sx1_1_tp; } ISV_Sx1_tp;</pre>	<pre>typedef struct tagISV_Sx1_0_tp {   Int16 Sx1_Gain;   Int16 Sx1_Gain1;   Int16 Sx2_Sum; } ISV_Sx1_0_tp;  typedef struct tagISV_Sx1_tp {   ISV_Sx1_0_tp * pISV_Sx1_0_tp; } ISV_Sx1_tp;</pre>
<b>Example 2</b>	
<pre>typedef struct tagISV_Sx1_1_tp {   Int16 Sx1_Gain1; } ISV_Sx1_1_tp;  typedef struct tagISV_Sx1_tp {   ISV_Sx1_1_tp * pISV_Sx1_1_tp; } ISV_Sx1_tp;</pre>	<pre>typedef struct tagISV_Sx1_0_tp {   Int16 Sx1_Gain1; } ISV_Sx1_0_tp;  typedef struct tagISV_Sx1_tp {   ISV_Sx1_0_tp * pISV_Sx1_0_tp; } ISV_Sx1_tp;</pre>



**Reasons**

- Readability
- Code size

**Migration issue** None

## MISRA Compliance

**Trigger-state-variable**

TargetLink no longer generates the state-variable associated with blocks whose trigger port is driven by a constant. The generated code is marked by the **condition for never trigger** code commentary.

TargetLink ≤ 4.2	TargetLink 4.3
Generated a state variable in non-optimized trigger code that was never read and removed by optimization.	Does not generate a state variable for trigger code.

**Reasons**

- Code efficiency
- MISRA C compliance

**Migration issue** None**Additional cast in ?-operation for min/max/abs in Stateflow code**

Calls of the **abs**, **min**, and **max** functions in Stateflow that do not have a parent assignment are generated with the **?** operator.

TargetLink now casts both of the operands to the result type of the function. The following code example is for an **abs** function:

TargetLink ≤ 4.2	TargetLink 4.3
<pre>(Ca14_I16In1 &gt;= 0) ? Ca14_I16In1 : ((UInt16) (-Ca14_I16In1))</pre>	<pre>(Ca14_I16In1 &gt;= 0) ? ((UInt16) (Ca14_I16In1)) : ((UInt16) (-Ca14_I16In1))</pre>

**Reason** MISRA C compliance**Migration issue** None**Arguments of fixed-point atan2()**

The arguments of the fixed-point **atan2()** function changed. The second parameter now always is of **Int32**, the third always **Int16**.

TargetLink ≤ 4.2
<pre>Aux_S16 = ((Int16) Sa1_YIn_Int8_4) * ((Int16) Sa1_YIn_Int8_4); F__I16ATAN2I16((Int16) Sa1_XIn_Int8_4, Aux_S16, Sa1_YIn_Int8_4)</pre>

**TargetLink 4.3**

```
Aux_S32 = (Int32) (((Int16) Sa1_YIn_Int8_4) * ((Int16) Sa1_YIn_Int8_4));
F__I16ATAN2I16((Int16) Sa1_XIn_Int8_4, Aux_S32, (Int16)Sa1_YIn_Int8_4)
```

**Reason** MISRA C compliance**Migration issue** None

## Mixed Operations (Floating-Point and Fixed-Point Types)

**Assignments in Stateflow**

Certain assignments might change in production code generated from Stateflow.

The following code is for an 8-bit integer variable called `Ca1_in` with `Min = 1` and `Max = 3`:

TargetLink ≤ 4.2	TargetLink 4.3
<b>Assignment of an integer to a floating-point</b>	
Calculation is in integer as long as possible to increase efficiency and precision:	
<code>Var = (Int16) (((Float64) (Int8) Ca1_in) + 1.);</code>	<code>Var = (Int16) (Float64) (Int8) (((Int8) Ca1_in) + 1);</code>
<b>Assignment of Float64 to Float32</b>	
Calculation is in Float64 as long as possible to increase precision:	
<code>F32Var = (Float32)F64Var + (Float32)F64Var2;</code>	<code>F32Var = (Float32)(F64Var + F64Var2);</code>

**Reason**

- Precision
- Code efficiency

**Migration issue** None**Non-integral values in Stateflow**

TargetLink now treats non-integral values in Stateflow as follows:

- Representable as `Float32` - TargetLink interprets the value as being a `Float32`
- Greater/smaller than the maximum/minimum of `Float32` - TargetLink interprets the value as being a `Float64`
- Cannot be precisely represented in `Float32` - TargetLink uses the `DefaultFloatType` as defined in `TargetConfig.xml`

This can result in code differences in conjunction with the **SupportSinglePrecisionLibraries** Code Generator option.

Additionally, non-integral values in an integer context result in floating-point operations:

TargetLink ≤ 4.2	TargetLink 4.3
Expression: I16UnscaledVar2 = sin(I16UnscaledVar1) + 0.5;	
Ca2_I16UnscaledVar2 = (Int16) (((Int16) (F__I16SINI16(Aux_) >> 14)) + 1)	Ca2_I16UnscaledVar2 = (Int16) (sinf((Float32) Ca2_I16UnscaledVar1) + 0.5F)
Expression: I16UnscaledVar2 = I16UnscaledVar1 + 0.5 + I16UnscaledVar2	
Ca2_I16UnscaledVar2 = (Int16) (((UInt16) (Ca2_I16UnscaledVar1 + 1)) + ((UInt16) Ca2_I16UnscaledVar2))	Ca2_I16UnscaledVar2 = (Int16) (((Float32) Ca2_I16UnscaledVar1) + 0.5F + ((Float32) Ca2_I16UnscaledVar2))

**Reason** Precision

**Migration issue** The production code contains floating-point operations.

**Solution** Do not use non-integral values in pure integer contexts.

**Math functions in Stateflow**

With TargetLink 4.3, the fixed-point implementation of math functions is called more frequently in an integer context in code generated from Stateflow, instead of float or double.

TargetLink ≤ 4.2
F64Var = (Float64) (Int8) (((Int8) sin(((Float64) I16Var) * 9.5874e-05)) << 3);
TargetLink 4.3
F64Var = (Float64) (Int8) (((Int8) (((Int32) F__I16SINI16(I16Var)) >> 14)) << 3);

**Reason** Code efficiency

**Migration issue** None

**Additional auxiliary variable for mod() in Stateflow**

The production code generated for Stateflow changed with respect to the mod() function if it is called with another operation as argument in a floating-point context. The operation in the argument is calculated only once and written to an auxiliary variable.

The following code is generated for  $\text{Out} = \text{mod}(\text{I32In} * \text{I32In} + \text{F64In}, 3)$ .

#### TargetLink ≤ 4.2

```
Out = (Int32) (((Float64) I32In) * ((Float64) I32In)) +
    F64In - (3. * ((Float64) (Int32) (((Float64) I32In) * ((Float64) I32In)) + F64In / 3.));
if (Out != 0) {
    if (((Float64) I32In) * ((Float64) I32In) + F64In < 0.) {
        Out += 3;
    }
}
```

#### TargetLink 4.3

```
Aux_ = (((Float64) I32In) * ((Float64) I32In)) + Sa1_In;
Out = (Int32) (Aux_ - (3. * ((Float64) (Int32) (Aux_ / 3.))));
if (Out != 0) {
    if (Aux_ < 0.) {
        Out += 3;
    }
}
```

**Reason** Code efficiency

**Migration issue** None

## State Reset

### State reset

For the call of the step function, a superfluous if statement (see the bold part in the following code example) is now avoided.

TargetLink ≤ 4.2	TargetLink 4.3
<pre>ConfigurationDeskif (Se1_InPort[1] &gt; 0.) {     if (!Se3_RSWE) {         INIT_Se3_subsystem2();         Se3_RSWE = 1;     }      Se3_subsystem2(); } else {     if (Se3_RSWE == 1) {         Se3_RSWE = 0;     } }</pre>	<pre>if (Se1_InPort[1] &gt; 0.) {     if (!Se3_RSWE) {         INIT_Se3_subsystem2();         Se3_RSWE = 1;     }      Se3_subsystem2(); } else {     Se3_RSWE = 0; }</pre>

**Reason** Suppress superfluous code

**Migration issue** None

### Chart triggering a subsystem/chart via a function-call output event bound to a state

A Stateflow chart be configured as follows:

- The chart is located in a conditional subsystem whose **States when enabling**, **States when action is resumed**, or **States when starting** property is set to **reset**.
- The chart triggers a subsystem or chart via a function-call output event.
- The output event is bound to a state of the triggering chart by means of a bind action.

The code changes are as follows:

- A Boolean **RSWE** flag for the function-call subsystem or Stateflow chart is generated.
- The **RSWE** flag is reset after the **INIT** function of the function-call subsystem or Stateflow chart was called in the state's entry function.
- The **INIT** function is called at the beginning of the state-during code. This call is encapsulated by a request to the **RSWE** flag.
- The **RSWE** flag can also be reset by other callers of the chart function, e.g., if the chart is located in an enabled subsystem.

#### Tip

By inlining the **INIT** function and the state's entry function, the (re-)setting of the **RSWE** flag as well as the state-reset code of the function-call subsystem or Stateflow chart can directly become part of the code.

TargetLink ≤ 4.2	TargetLink 4.3
<pre> void Enabled(void) {     if (Sa1_InPort &gt; 0) {         if (!(Sa2_RSWE)) {             INIT_Sa2_Enabled();              Sa2_RSWE = 1;         }          Sa2_Enabled();     }     else {         Sa2_RSWE = 0;     }     [...] } </pre>	<pre> void Enabled(void) {     if (Sa1_InPort &gt; 0) {         if (!(Sa2_RSWE)) {             INIT_Sa2_Enabled();              Sa2_RSWE = 1;         }          Sa2_Enabled();     }     else {         Sa2_RSWE = 0;          Sa1_RSWE = 0;     }     [...] } </pre>

TargetLink ≤ 4.2	TargetLink 4.3
<pre> void Ca2_Calling(void) {     if (SIBFS_Calling_a.Ca3_B == 1) {         if (Ca2_counter == 5.) {             [...]             Ca4_A_en();         }         [...]     }     else {         if (SIBFS_Calling_a.Ca4_A == 1) {              [...]         }         else {             Ca4_A_en();         }     } }  void Ca4_A_en(void) {     [...]      INIT_Ca1_Chart(); } </pre>	<pre> void Ca2_Calling(void) {     if (SIBFS_Calling_a.Ca3_B == 1) {         if (Ca2_counter == 5.) {             [...]             Ca4_A_en();         }         [...]     }     else {         if (SIBFS_Calling_a.Ca4_A == 1) {             if (!(Sa1_RSWE)) {                 INIT_Ca1_Chart();                 Sa1_RSWE = 1;             }              [...]         }         else {             Ca4_A_en();         }     } }  void Ca4_A_en(void) {     [...]      INIT_Ca1_Chart();     Sa1_RSWE = 1; } </pre>

**Reason** Resolve differences in MIL/SIL/PIL simulation modes

**Migration issue** None

### Chart triggering a referenced subsystem via a function-call output event

A Stateflow chart be configured as follows:

- The chart triggers a referenced subsystem via a function-call output event.
- The output event is not bound to any state of the triggering chart.
- For the referenced subsystem, the trigger port's **States when enabling** property is set to **reset**.

The code changes are as follows:

- A Boolean **RSWE** flag for the function-call subsystem is generated.
- The Stateflow chart function contains the following two statements at its beginning, both of which are encapsulated by a request to the **RSWE** flag:
  1. Call of the function-call subsystem's **INIT** function
  2. Reset of the **RSWE** flag

TargetLink ≤ 4.2	TargetLink 4.3
<pre>void Ca1_Chart(void) {    /* Begin execution of chart &lt;...&gt; */   ...   /* End execution of chart &lt;...&gt; */ }</pre>	<pre>void Ca1_Chart(void) {   if (!(SMR1_RSWE)) {     /* initialization of subsystem: &lt;...&gt; */     INIT_SMR1();     SMR1_RSWE = 1;   }    /* Begin execution of chart &lt;...&gt; */   ...   /* End execution of chart &lt;...&gt; */ }</pre>

In combination with the code change **Init of function-call-triggered referenced models** described in [Other](#) on page 250, as well as additional code optimization and function inlining, the code changes can look like this:

<pre>void sr_held_mr_reset_forced(void) {   /* SLStaticLocalInit: Default storage class for static local variables with initvalue   Width: 8 */   static Bool Sa2_RSWE = 0;    /* sr_held_mr_reset_forced/Subsystem/Enable: Enable condition */   if (Sa1_in_ &gt; 0) {     ...      if (!(Sa2_RSWE)) {       /* initialization of subsystem: sr_held_mr_reset_forced/Subsystem/Model */       INIT_SMR1();        /* set system state to 'enabled' */       Sa2_RSWE = 1;     }      ...   }   else {     /* set system state to 'disabled': sr_held_mr_reset_forced/Subsystem */     Sa2_RSWE = 0;   } }</pre>	<pre>void sr_held_mr_reset_forced(void) {   /* SLStaticLocalInit: Default storage class for static local variables with initvalue   Width: 8 */   static Bool SMR1_RSWE = 0;    /* sr_held_mr_reset_forced/Subsystem/Enable: Enable condition */   if (Sa1_in_ &gt; 0) {     ...      if (!(SMR1_RSWE)) {       /* initialization of subsystem: sr_held_mr_reset_forced/Subsystem/Model */       INIT_SMR1();        /* set system state to 'enabled' */       SMR1_RSWE = 1;     }      ...   }   else {     /* set system state to 'disabled': sr_held_mr_reset_forced/Subsystem */     SMR1_RSWE = 0;   } }</pre>
--	--

**Reason** Resolve differences in MIL/SIL/PIL simulation modes

**Migration issue** None

**State reset for flip flop blocks** The code generated for the following blocks changed:

- **D Flip-Flop** blocks
- **J-K Flip-Flop** blocks

TargetLink ≤ 4.2	TargetLink 4.3
Scope changes for J_Previous, K_Previous, and D_Previous variables:	
The variables had local scope with static storage duration.	The variables have global scope unless otherwise specified.
State reset of the J_Previous, K_Previous, and D_Previous variables:	
No state reset.	State reset in the <b>INIT</b> function of the parenting subsystem. The <b>INIT</b> function is generated only if the parenting subsystem or one of its callers are configured for state reset. Usually, this function is inlined during optimization. The change in the variable's scope changes the context of the variables. This might cause changes in variable names because the <b>\$R</b> name macro expands differently.

**Reason** Resolve differences in MIL/SIL/PIL simulation modes

**Migration issue** None

**State reset for actual parameters of latched inports of triggered subsystems**

The code for latched inport blocks of triggered subsystems changed if their triggered subsystems are parented by a subsystem that is configured for state reset. The state variables are now of global scope.

TargetLink ≤ 4.2	TargetLink 4.3
State reset of variables:	
No state reset.	State reset in the <b>INIT</b> function of the parenting subsystem. The <b>INIT</b> function is generated only if the parenting subsystem or one of its callers are configured for state reset. Usually, this function is inlined during optimization. The change in the variable's scope changes the context of the variables. This might cause changes in variable names because the <b>\$R</b> name macro expands differently.

**Reason** Resolve differences in MIL/SIL/PIL simulation modes

**Migration issue** None

**INIT function for referenced models**

The code changed for referenced models that are configured as follows:

- Contains a Trigger block with **Trigger type** set to **function-call** and **States when enabling** set to **held**
- Contains a **Function** block at its root level whose **forceinitfunction** block property is set to **on**

TargetLink ≤ 4.2	TargetLink 4.3
Generate an <b>INIT</b> function for a state reset that is never called.	Generate an empty <b>INIT</b> function that is called depending on the modeling of the subsystem that contains the Model block.

**Reason** Suppress superfluous code

**Migration issue** To force TargetLink to generate the same **INIT** function as in previous versions, set the Trigger block's **States when enabling** property to **reset**. TargetLink then generates and calls the **INIT** function that contains the reset code.



**Init of function-call triggered referenced models**

With the improved support for held/reset the code changed for referenced models, if the following conditions are fulfilled:

- The Model block references a model that contains a Trigger block with **Trigger type** set to **function-call** and **States when enabling** set to **reset**.
- The Model block is triggered by a subsystem whose states are hold.

TargetLink ≤ 4.2	TargetLink 4.3
<p>The <b>INIT</b> function generated for the calling subsystem simply called the <b>INIT</b> function of the referenced model.</p> <pre> void Sa2_Subsystem(void) {     /* Subsystem/Subsystem/Function-Call Generator     call of function: Subsystem/MRSystem1 */     MR_STEP(); } void INIT_Sa2_Subsystem(void) {     /* initialization of subsystem: Subsystem/MRSystem1 */     MR_INIT(); }                     </pre>	<p>The <b>INIT</b> function generated for the referenced model is called in the step function of the calling subsystem. The call of the <b>INIT</b> function is placed before the call of the step function that is generated for the referenced model. Additionally, a <b>RSWE</b> flag is generated to control whether the <b>INIT</b> function is called.</p> <pre> void Sa2_Subsystem(void) {     if (!(SMR11_RSWE)) {         /* initialization of subsystem: Subsystem/MRSystem1 */         MR_INIT();         SMR11_RSWE = 1;     }     /* Subsystem/Subsystem/Function-Call Generator     call of function: Subsystem/MRSystem1 */     MR_STEP(); }                     </pre>

**Reason** Increase consistency

**Migration issue** None

**Unconditionally executed subsystems and function-call-triggered references models**

The code has changed for certain subsystems, calling referenced models if the following conditions are fulfilled:

- The calling subsystem is unconditional:
  - It is not enabled.
  - It is not triggered.
  - It is not action-triggered.
  - It is not triggered-and-enabled.
  - It is not iterated.
- The referenced model is specified as follows:
  - Contains a Trigger block with **Trigger type** set to **function-call** and **States when enabling** set to **reset**

TargetLink ≤ 4.2	TargetLink 4.3
<p>The <b>INIT</b> function of the referenced model was not called in the generated code. Message <b>W15371</b> was displayed.</p>	<p>The <b>INIT</b> function of the referenced model now is called in the generated code.</p>

**Reason** Resolve differences in MIL/SIL/PIL simulation modes

**Migration issue** None

## Related topics

### Basics

[Basics on Resettable Subsystems](#) (📖 [TargetLink Customization and Optimization Guide](#))

## Other

### Sorting of internal default variable classes

TargetLink sorts internal default variable classes below those loaded from the TargetLink Data Dictionary.

This sorting now also works for the following elements:

- Default variable classes derived with a new **Type** prefix from an existing one, e.g., in case of indirect function reuse.
- Width-specific DD **VariableClass** templates.

#### TargetLink ≤ 4.2

```

/*****\
  SLGlobal: Default storage class for global variables | Width: 64
\*****/
Float64 Sa1_InPort;
  5 /* Sr1_Gain_gain */
};
/*****\
  UserSLFcnOutput: SLFcnOutput = { default GLOBAL default default default } | Width: 64
\*****/
Float64 Sa1_OutPort; /* Considered default because UserSLFcnOutput is changed to GLOBAL via template only for
Width = Bitfield */
/*****\
  UserSLFcnOutput_a: Derived TL_CG default variable class. | Width: 32
\*****/
ISV_Sr1_tp * pISVsr1; /* Considered default because UserSLFcnOutput is changed to GLOBAL via template only for
Width = Bitfield */

```

**TargetLink 4.3**

```

/*****\
  UserSLFcnOutput: SLFcnOutput = { GLOBAL } | Width: 64
\*****/
Float64 Sa1_OutPort; /* Not considered default because UserSLFcnOutput is changed to GLOBAL via template */
/*****\
  UserSLFcnOutput_a: Derived TL_CG default variable class. | Width: N.A.
\*****/
ISV_Sr1_tp * pISVsr1; /* Not considered default because UserSLFcnOutput is changed to GLOBAL via template */
/*****\
  SLGlobal: Default storage class for global variables | Width: 64
\*****/
Float64 Sa1_InPort;
    5 /* Sr1_Gain_gain */
};

```

**Reason**    Increase consistency

**Migration issue**    None

**Changed code comments**

Several of TargetLink's code comments changed:

- Declaration comments for enum and pointer variables now have **Width: N.A.**
- Comments in the code pattern associated with non-scalar interrunnable variables, or `NvDataElements` are sorted differently.
- Comments associated with function headers, code sections, or preprocessor control flow can now contain additional whitespaces.
- Comments for the structured data types of sub reuse structs changed from `/* Description: Reuse linker section structure */` to `/* Description: Reuse substructure */`.
- Comments generated for blocks of declarations and definitions of functions and variables are no longer limited to functions or variables with the same **FunctionClass** or **VariableClass**, resulting in fewer, larger blocks.

**Reasons**

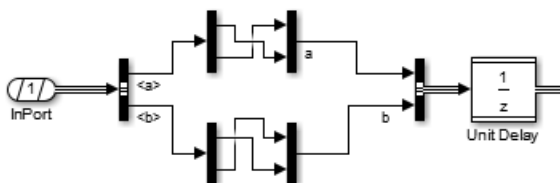
- Increase consistency
- Readability

**Migration issue**    None

**Inheritance of structured data types**

The inheritance of structured data types changed. For model and code consistency, bus-capable simulation blocks now inherit structured data types only if the bus signal that connects them to the inheriting block is not modified along the signal line.

The following code is for a **Unit Delay** block that is contained in a model that looks like this:



TargetLink ≤ 4.2	TargetLink 4.3
<pre> struct BS_IP_Sa1_InPort Sa1_InPort; struct BS_IP_Sa1_InPort Sa1_OutPort;  struct BS_IP_Sa1_InPort X_Sa1_Unit_Delay = {     {         0, 0     },     {         0, 0     } };  void TL_Root(void) {     /* BusOutputport: TL_Root/OutPort */     Sa1_OutPort = X_Sa1_Unit_Delay;      /* Unit delay: TL_Root/Unit Delay */     X_Sa1_Unit_Delay.Sa1_a[0] = Sa1_InPort.Sa1_a[1];     X_Sa1_Unit_Delay.Sa1_a[1] = Sa1_InPort.Sa1_a[0];     X_Sa1_Unit_Delay.Sa1_b[0] = Sa1_InPort.Sa1_b[1];     X_Sa1_Unit_Delay.Sa1_b[1] = Sa1_InPort.Sa1_b[0]; } </pre>	<pre> struct BS_IP_Sa1_InPort Sa1_InPort; struct BS_IP_Sa1_InPort Sa1_OutPort;  Int16 X_Sa1_Unit_Delay[2] = {     0, 0 };  Int16 X_Sa1_Unit_Delay_a[2] = {     0, 0 };  void TL_Root(void) {     /* BusOutputport: TL_Root/OutPort */     Sa1_OutPort.Sa1_a[0] = X_Sa1_Unit_Delay[0];     Sa1_OutPort.Sa1_a[1] = X_Sa1_Unit_Delay[1];     Sa1_OutPort.Sa1_b[0] = X_Sa1_Unit_Delay_a[0];     Sa1_OutPort.Sa1_b[1] = X_Sa1_Unit_Delay_a[1];      /* Unit delay: TL_Root/Unit Delay */     X_Sa1_Unit_Delay[0] = Sa1_InPort.Sa1_a[1];     X_Sa1_Unit_Delay[1] = Sa1_InPort.Sa1_a[0];     X_Sa1_Unit_Delay_a[0] = Sa1_InPort.Sa1_b[1];     X_Sa1_Unit_Delay_a[1] = Sa1_InPort.Sa1_b[0]; } </pre>

**Reason** Bug fix

**Migration issue** In TargetLink 4.2, the bus signal could be modified as follows without impeding inheritance:

- Changing the order of bus elements via Selector, Demux, or Mux blocks
- Transposing matrices via Permute Dimensions blocks

This is no longer possible in TargetLink 4.3. If you need inheritance, adjust your models accordingly.

**Solution** Explicitly specify a structured data type at the inheriting block.

### Sorting of variable definitions

TargetLink now sorts the definitions of certain variables more consistently. This applies to the following contexts:

- Variables associated with variable classes that were derived from internal default variable classes

- Type prefixes
- TOMs
- Initialized variables that are relevant for the variable vector width and are declared between preprocessor directives.
- Scope reduction of variables to function-local scope that initially had a variable class with empty **ScopeReducedClass** property

**Reason** Increase consistency

**Migration issue** None

**Code generated from model-based CGUs**

The code generated from modules whose **ModuleInfo.CodeGenerationBasis** property is set to **ModelBased** changed, if the CGU from which the code was generated is not the specific owner of the module:

TargetLink ≤ 4.2	TargetLink 4.3
The module was treated as if its <b>ModuleInfo.CodeGenerationBasis</b> property was set to <b>ModelAndDDBased</b> : it contained DD-based code that was not used in the model by the current CGU.	The model is now treated as if its <b>ModuleInfo.CodeGenerationBasis</b> property is set to <b>ModelBased</b> : it now only contains code belonging to the model. DD-based code that is not used in the model by the current CGU is no longer generated in the module.

**Reason** Conform with user expectations

**Migration issue** To generate code that matches the code of prior TargetLink versions, set the module's **ModuleInfo.CodeGenerationBasis** property to **ModelAndDDBased**.

**VariableTemplates with VariableKind GlobalInterfaceVar**

DD **VariableTemplate** objects whose **VariableKind** property is set to **GlobalInterfaceVar** and whose **WidthSpec** property is not set or empty now also apply to implicitly generated interface variables with a structured data type.

The following code example shows an implicitly generated global interface variable. It results from a variable template that is specified as follows:

- **Filter.VariableKind** = **GlobalInterfaceVar**
- **Filter.WidthSpec** = <value not set>
- **Filter.VariableClassSpec** = <value not set>
- **Settings.NameTemplate** = **GLB\_IF\_VAR\_\$\$R**
- **Settings.VariableClass** = <value not set>
- **Settings.Type** = <value not set>

TargetLink ≤ 4.2	TargetLink 4.3
<pre>struct BS_IP_Sa2_In1 IF_Sa2_Out1 = {     0, /* Sa2_a */     0 /* Sa2_b */ }</pre>	<pre>struct BS_IP_Sa2_In1 GLB_IF_VAR_Out1 = {     0, /* Sa2_a */     0 /* Sa2_b */ };</pre>

**Reason** Conform with user expectations

**Migration issue** If you want to restrict the **VariableTemplate** to plain variables, select all the bit widths in the **Filter.WidthSpec** property's bitfield.

**Code optimization and Stateflow**

TargetLink no longer partially optimizes the code generated from Stateflow if code optimization is disabled. This affects the propagation of initial values and folding of arithmetic operations. Additionally, if optimization is enabled, suitable code comments are generated.

**Reasons**

- Conform with user expectations
- Readability

**Migration issue** None

**2-D look-up table functions with matrix parameters**

The code changed for 2-D look-up table functions with matrix parameters:

TargetLink ≤ 4.2	TargetLink 4.3
Access to the matrix itself:  <pre>int tab[3][4]; int axis[6];  fcn(&amp;(axis[0]), N, x);  fcn2(tab, N, x, y);</pre>	Access to the matrix's first element address:  <pre>int tab[3][4]; int axis[6];  fcn(&amp;(axis[0]), N, x);  fcn2(&amp;(tab[0][0]), N, x, y);</pre>

**Reason** Bug fix for access functions

**Migration issue** None

**Addition and subtraction with saturation**

The code might change for additions and subtractions with saturation, if the **ExploitComputeThroughOverflow** Code Generator option is set to **1 - Never**.

```
TargetLink ≤ 4.2

Sa1_I32ADDI32I32 = Sa1_InPort_Int32 + Sa1_InPort1_Int32;
if ((Sa1_InPort_Int32 >= 0) && (Sa1_InPort1_Int32 >= 0) && (Sa1_I32ADDI32I32 < 0)) {
    Sa1_I32ADDI32I32 = 2147483647;
}
else {
    if ((Sa1_InPort_Int32 < 0) && (Sa1_InPort1_Int32 < 0) && (Sa1_I32ADDI32I32 >= 0)) {
        Sa1_I32ADDI32I32 = (-2147483647L -1L) /* INT32MIN */;
    }
}
```

**TargetLink 4.3**

```

if ((Sa1_InPort1_Int32 > 0) && (Sa1_InPort_Int32 > (2147483647 - Sa1_InPort1_Int32))) {
    Sa1_I32ADDI32I32 = 2147483647;
}
else {
    if ((Sa1_InPort1_Int32 < 0) && (Sa1_InPort_Int32 < ((-2147483647L -1L) /* INT32MIN */ /* -2147
483648. */ - Sa1_InPort1_Int32))) {
        Sa1_I32ADDI32I32 = (-2147483647L -1L) /* INT32MIN */;
    }
    else {
        Sa1_I32ADDI32I32 = Sa1_InPort_Int32 + Sa1_InPort1_Int32;
    }
}
}

```

**Reason** Conform with user expectations

**Migration issue** None

## Discontinuations

**Where to go from here**

**Information in this section**

<a href="#">Discontinued TargetLink Features .....</a>	<a href="#">255</a>
<a href="#">Obsolete API Functions .....</a>	<a href="#">256</a>
<a href="#">Obsolete Limitations .....</a>	<a href="#">256</a>

## Discontinued TargetLink Features

**MISRA C:2004 Compliance Documentation document**

As announced previously, the MISRA C:2004 Compliance Documentation document was discontinued. Use the [MISRA C:2012 Compliance Documentation](#) document together with [MISRA C:2012 Addendum 1 - Rule mapping](#) instead.

## Obsolete API Functions

### List of obsolete API functions

Function	Status	Replacement
<code>t1_distribute_refmodel_files</code>	Error <sup>1)</sup>	_2)
<code>t1_integrate_refmodel_files</code>	Error <sup>1)</sup>	_2)
<code>t1_propman</code>	Warning <sup>3)</sup>	<code>t1Propman</code>

<sup>1)</sup> The function was removed from TargetLink.

<sup>2)</sup> Specify artifact locations in the Data Dictionary instead. Refer to [Specifying the Location of Artifacts Generated or Used by TargetLink](#) ([TargetLink Customization and Optimization Guide](#)).

<sup>3)</sup> The function is obsolete and will be removed in a future version of TargetLink.

**Compatibility consideration**    Adapt your user scripts and tool chain accordingly.

## Obsolete Limitations

**Obsolete with TargetLink 4.3**    With TargetLink 4.3, the following limitations of previous TargetLink versions were removed:

### AUTOSAR limitations

#### Unsupported calls to RTE API functions

TargetLink does not provide native support for the following RTE API function:

- `Rte_IsUpdated`

### Block-specific limitations

#### Inheritance of block properties

The Code Generator does not support any Min/Max constraints at the input and output signals of the following blocks:

- **Bit Set**
- **Bit Clear**
- **Bitwise Operator**
- **Shift Arithmetic**

#### Logical Operator block

TargetLink does not support the NXOR operation.



---

## Stateflow Limitations

### State reset behavior of function-called charts

TargetLink supports only the setting **inherit** for the state reset behavior of function-called charts.

---

## Subsystem Creation Limitations

### States in function-call-triggered subsystems

MATLAB/Simulink provides the **States when enabling** property for the Trigger block when the function-call trigger type is selected. It lets you influence the way states are dealt with in function-call-triggered subsystems. For function-call-triggered subsystems, TargetLink does not support the **held** and **reset** settings of the **States when enabling** property (this limitation does not apply to the root level of referenced models). To avoid problems with function-call-triggered subsystems, select **inherit**.

## Changes in Future TargetLink Versions

### Where to go from here

### Information in this section

Features to Be Discontinued .....	258
API Functions to Be Discontinued .....	259
Deprecated Code Generator Options .....	260
Other .....	260

## Features to Be Discontinued

### Clean code and Do not log anything

Variables selected for logging cannot be fully optimized. When generating code with the **Global logging option Do not log anything** or **Log according to block data**, TargetLink does not fully optimize the code to facilitate testing. That is, the code only differs with regard to the log macros. This contrasts the **Clean code** checkbox on the **Code Generation** page of the **TargetLink Main Dialog** block, which always activates full code optimization.

The special **Do not log anything** behavior will be removed in future TargetLink versions.

### User state flags in Stateflow

Support for TargetLink's own user state flags feature in Stateflow will be discontinued in future TargetLink versions, because the Stateflow Active State data is similar and more convenient to use.

### Simulink's classic initialization mode

Support for Simulink's [classic initialization mode](#) ([Glossary](#)) will be discontinued in future TargetLink versions.

### Dynamic components

Support for specifying dynamic components for DD **Variable** objects will be discontinued in future TargetLink versions.

### Code generation for special OSEK versions

The code generation for special OSEK versions, such as OsCan, will be discontinued in future TargetLink versions.

### Signal logging format

Support for Simulink's logging method **Mode1DataLogs (Signal logging format parameter)** will be discontinued in future TargetLink versions.

<b>Automatic interpretation of Boolean</b>	TargetLink's automatic interpretation of certain integer data types as Boolean will be discontinued in future TargetLink versions.
<b>Support of AUTOSAR 2.x and 3.x</b>	TargetLink's support of AUTOSAR 2.x and AUTOSAR 3.x will be discontinued in future TargetLink versions.
<b>Unit Delay Reset Enabled</b>	TargetLink's support of the <b>Unit Delay Reset Enabled</b> block will be discontinued in future TargetLink versions.

## API Functions to Be Discontinued

**Discontinued API functions** The following API functions are deprecated and will be removed in future TargetLink versions:

Function	Deprecated Since	Replacement Function
t1_adapt_dd_references	TargetLink 4.0	t1MoveDDObject
t1_extract_subsystem	TargetLink 4.0	t1ExtractSubsystem
t1_find_dd_references	TargetLink 4.0	t1FindDDReferences
t1_get_blockset_mode	TargetLink 4.0	t1OperationMode
t1_sim_interface	TargetLink 4.0	t1SimInterface
t1_switch_blockset	TargetLink 4.0	t1OperationMode
t1_upgrade	TargetLink 4.0	t1Upgrade
generate_ASAP2	TargetLink 2.x	dsdd_export_a2l_file
t1_upgrade_libmapfile	TargetLink 4.0	-

### Note

See the help contents on the new API functions to adjust your user scripts accordingly.

### Related topics

#### References

[t1SimInterface](#) (📖 TargetLink API Reference)

## Deprecated Code Generator Options

---

### List of deprecated Code Generator options

The following Code Generator options are deprecated and will be removed in future TargetLink versions:

- [SideEffectFreeAnalysisThreshold](#) ([📖 TargetLink Block and Object Reference](#))
- [TreatAllForcedAtomicSubsystemsAsWeakAtomic](#) ([📖 TargetLink Block and Object Reference](#))
- [DisableFunctionsAsAnalysisBoundaries](#) ([📖 TargetLink Block and Object Reference](#))
- [CreateRestartFunctions](#) ([📖 TargetLink Block and Object Reference](#))

## Other

---

### Encoding of A2L files

Generated A2L files will be encoded in UTF-8 with BOM in a future TargetLink version (as recommended by newer ASAM MCD-2 MC standard versions).

# VEOS

## Where to go from here

## Information in this section

<a href="#">New Features of VEOS 4.1</a> .....	261
Gives an overview of the new features of VEOS 4.1.	
<a href="#">Compatibility of VEOS 4.1</a> .....	262
Provides information on the compatibility of VEOS 4.1.	
<a href="#">Migrating to VEOS 4.1</a> .....	264
<a href="#">Discontinuations in VEOS</a> .....	265

## New Features of VEOS 4.1

### **New licensing for dSPACE products**

With dSPACE Release 2017-B, dSPACE introduces a new licensing technology for protecting dSPACE software. dSPACE has also improved the software installation process. As a consequence, some parts of the dSPACE software, for example, the user documentation, are installed in encrypted archives on the host PC. These license-protected archives must be decrypted before you can use the contained files, for example, to view the user documentation. Refer to [New Licensing for dSPACE Products](#) on page 33.

### **Import of SMC files created with SYNECT**


VEOS Player now supports the import of system model container (SMC) files created with SYNECT.

For instructions, refer to [How to Import System Model Containers](#) ( [VEOS Manual](#)).

**Supported compiler versions**

VEOS 4.1 supports the following compiler versions:


- GCC 5.2 (newly supported)
- MSVC 11
- MSVC 14 (newly supported)

For details, refer to [Basics on Integrating the Simulation System](#) ( [VEOS Manual](#)).

**Undoing/Redoing the most recent actions**

VEOS Player now lets you undo/redo the most recent commands or actions.


Refer to:

- [Undo](#) ( [VEOS Manual](#))
- [Redo](#) ( [VEOS Manual](#))

**Customizing the screen arrangement**

You can now customize the screen arrangement of VEOS Player.

The screen arrangement defines which controlbars are displayed and how they are arranged. The first time you execute the application, it starts with a default screen arrangement that you can modify.

For instructions, refer to [How to Customize the Screen Arrangement](#) ( [VEOS Manual](#)).

## Compatibility of VEOS 4.1


**Where to go from here**

**Information in this topic**

<a href="#">Compatibility overview</a> .....	262
<a href="#">Compatibility in general</a> .....	262
<a href="#">Supported compiler versions</a> .....	262
<a href="#">BSC compatibility</a> .....	263
<a href="#">CTLGZ compatibility</a> .....	263
<a href="#">FMU compatibility</a> .....	263
<a href="#">OSA compatibility</a> .....	263
<a href="#">Real-Time Testing compatibility</a> .....	263
<a href="#">SIC compatibility</a> .....	264
<a href="#">SMC compatibility</a> .....	264

**Compatibility overview**

**Compatibility in general** dSPACE recommends using only software products from the same dSPACE Release. This ensures maximum run-time compatibility.

**Supported compiler versions** For information on supported compiler versions, refer to [Basics on Integrating the Simulation System](#) ( [VEOS Manual](#)).

**BSC compatibility** VEOS 4.1 is compatible with bus simulation container (BSC) files created with the Bus Manager of dSPACE Release 2017-B (BSC version 1.3).

**CTLGZ compatibility** The following table shows the compatibility between VEOS 4.1 and CTLGZ files (V-ECU implementations):

V-ECU Implementations Created With...	V-ECU Implementation Version
dSPACE Release 2017-B: <ul style="list-style-type: none"> <li>SystemDesk 5.0</li> <li>TargetLink 4.3</li> </ul>	2.6
dSPACE Release 2017-A: <ul style="list-style-type: none"> <li>SystemDesk 4.8</li> </ul>	2.5
dSPACE Release 2016-B: <ul style="list-style-type: none"> <li>SystemDesk 4.7</li> <li>TargetLink 4.2</li> </ul>	2.4.1
dSPACE Release 2016-A: <ul style="list-style-type: none"> <li>SystemDesk 4.6</li> </ul>	2.4

**FMU compatibility** VEOS supports:

- Functional Mock-up Units (FMUs) that comply with the FMI 2.0 standard
- Only the FMI for Co-Simulation interface, but not the FMI for Model Exchange interface

For detailed and up-to-date compatibility information on dSPACE's FMI support, refer to:

<http://www.dspace.com/go/FMI-Compatibility>.

**OSA compatibility** The following table shows the compatibility between VEOS 4.1 and offline simulation application (OSA) files:

OSA Files Created with Products Of ...	OSA Version
dSPACE Release 2017-B	4.1 <sup>1)</sup>
dSPACE Release 2017-A	4.0 <sup>2)</sup>
dSPACE Release 2016-B	3.7 <sup>2)</sup>
dSPACE Release 2016-A	3.6 <sup>2)</sup>

<sup>1)</sup> OSA files created or modified with VEOS 4.1 cannot be loaded in earlier VEOS versions.

<sup>2)</sup> You cannot modify the properties of VPUs contained in an OSA file if you open the OSA file in a later VEOS version than the version with which the OSA file was originally created. However, port and network connections can be edited. As a consequence, it is recommended to rebuild the binary OSA files from existing model implementation container files (CTLGZ, SIC, BSC, FMU) when you migrate from one VEOS version to another.

**Real-Time Testing compatibility** To use RTT in connection with VEOS and ControlDesk, the Real-Time Testing (RTT) version used by the VEOS Simulator that runs the simulation system and the RTT version that is active on the PC must be identical.

The following table shows the VEOS Simulator version and the corresponding RTT version:

VEOS Simulator	Real-Time Testing Version
... from VEOS 4.1	Real-Time Testing Version 3.3
... from VEOS 4.0	Real-Time Testing Version 3.2
... from VEOS 3.7	Real-Time Testing Version 3.1
... from VEOS 3.6	Real-Time Testing Version 3.0

ControlDesk 6.2 automatically uses the VEOS Simulator from VEOS 4.1. You can therefore use RTT in connection with VEOS and ControlDesk if RTT 3.3 is active on the PC.

**SIC compatibility** The following table shows the compatibility between VEOS 4.1 and Simulink implementation container (SIC) files:

SIC Files Created with Model Interface Package for Simulink of ...	SIC Version
dSPACE Release 2017-B (Model Interface Package for Simulink 3.5)	1.3
dSPACE Release 2017-A (Model Interface Package for Simulink 3.4)	1.2.1
dSPACE Release 2016-B (Model Interface Package for Simulink 3.3)	1.2
dSPACE Release 2016-A (Model Interface Package for Simulink 3.2)	1.1

**SMC compatibility** VEOS 4.1 is compatible with system model container (SMC) files created with SYNECT 2.4 of dSPACE Release 2017-B (SMC version 1.0).

## Migrating to VEOS 4.1

### Introduction

To migrate from VEOS 4.0 to VEOS 4.1, you might have to carry out the following migration steps.

#### Note

To migrate to VEOS 4.1 from versions earlier than 4.0, you might also have to perform the migration steps of the intervening VEOS versions.

### Changed behavior when importing elements to VEOS Player via automation

### Changed behavior when importing elements to VEOS Player via its user interface

- Up to and including Version 4.0, VEOS Player automatically saved the project after you imported, for example, a V-ECU implementation using the **Import** command.



- As of Version 4.1, VEOS Player does not automatically save the project after you use the **Import** command.  
Use the **Save** command after the import.

**Changed behavior when importing elements to VEOS Player via automation** The behavior of the following methods of the **IProject** interface was changed in VEOS 4.1:

- **Import**
- **ImportNamedVpu**
- **ImportVpus**

Using these methods, you can import container files and OSA files to the project that is open in VEOS Player.

- Up to and including Version 4.0, VEOS Player automatically saved the project after you used one of the **Import** methods listed above.
- As of Version 4.1, VEOS Player does not automatically save the project after you use one of the **Import** methods listed above.

Use the **Save** method of the **IProject** interface after the import.

Refer to **IProject** ([📖 VEOS Manual](#)).

---

### Migrating from prior VEOS versions

To migrate from prior VEOS versions and reuse existing offline simulation applications, you might have to carry out additional migration steps. For more information on the migration steps, refer to [Migrating from Prior Versions of VEOS](#) ([📖 VEOS Manual](#)).

## Discontinuations in VEOS

---

### Discontinuations as of VEOS 4.1

**Discontinuation of the import of CTLGZ 2.0 files** As of dSPACE Release 2017-B, VEOS no longer supports the import of version 2.0 V-ECU implementation (CTLGZ) files.



# Compatibility Information

---

## Where to go from here

## Information in this section

Supported MATLAB Releases .....	267
Operating System .....	268
Notes on 64-Bit Compatibility of dSPACE Products .....	270
Run-Time Compatibility of dSPACE Software .....	270
Limitations for Using Windows Features .....	271

## Supported MATLAB Releases

---

### MATLAB®

Working with various dSPACE products requires that MATLAB is installed on your host PC.

#### Tip

For system requirements of MathWorks® software, refer to <http://www.mathworks.com/support/sysreq.html>.

MATLAB Release...	...Is Supported by dSPACE Release 2017-B					
	RCP and HIL Software <sup>1)</sup>	AutomationDesk 5.5 <sup>2)</sup>	TargetLink 4.3	Model Compare 2.8	dSPACE Python Extensions 2.4 <sup>3)</sup>	XIL API .NET MAPort 2017-B
R2017b (64-bit)	✓ <sup>4)</sup>	✓	✓	✓	✓	✓
R2017a (64-bit)	✓	✓	✓	✓	✓	✓
R2016b (64-bit)	✓	✓	✓	✓	✓	✓
R2016a (64-bit)	✓	✓	✓	✓	✓	✓

- <sup>1)</sup> 'RCP and HIL software' is a generic term for a software package containing several dSPACE software products, for example RTI, ConfigurationDesk, MotionDesk and ModelDesk. These software products are installed in a common folder.
- <sup>2)</sup> AutomationDesk's MATLAB Access library requires MATLAB.
- <sup>3)</sup> matlablib2 of dSPACE Python Extensions requires MATLAB.
- <sup>4)</sup> R2017b is not supported by the RTI FPGA Programming Blockset – FPGA Interface.

For up-to-date information on additional MATLAB releases that can be used in combination with dSPACE software, refer to <http://www.dspace.com/go/MATLABCompatibility>.

**Limitations for ModelDesk** If you use MATLAB R2016a for Simulink simulation and the simulation is running, a download requires at least one minute.

There is no limitation when the simulation is stopped or paused.

## Operating System

### Operating system on host PC

The dSPACE products of dSPACE Release 2017-B support the following operating systems:

- Windows 7 Professional, Ultimate, and Enterprise with Service Pack 1 (64-bit versions)  
Only the listed editions are supported. The Windows 7 Home and Starter editions are not supported.
- The following editions and servicing options of Windows 10:
  - Windows 10 Professional, Education, and Enterprise (64-bit versions)

The Windows 10 Home, Mobile, and Windows 10 S editions are not supported.

- Long-Term Servicing Branch: LTSB 2016
- Current Branch (CB) and Current Branch for Business (CBB): The compatibility statement of Microsoft applies that newer versions of these branches should be compatible with all previous version. dSPACE used CB 1703 for testing.

Some limitations apply when you use dSPACE software in conjunction with features of Windows. Refer to [Limitations for Using Windows Features](#) on page 271.

---

### Using MicroAutoBox Embedded PC as host PC

ControlDesk can also be installed on MicroAutoBox Embedded PC with 3rd Gen. Intel® Core™ i7-3517UE Processor running on Windows 7 Professional, Ultimate, and Enterprise, 64-bit version.

---

### Allowing communication

**Installing of additional firewall rules** Additional Windows firewall rules are installed during the installation of various dSPACE software products. For example, one rule allows communication with a dSPACE expansion box, such as AutoBox. Another rule allows MotionDesk to receive motion data from a network channel. These example rules are created by the following commands:

- `netsh advfirewall firewall add rule name="dSPACE Net Service" service=any dir=in action=allow profile=any protocol=icmpv4:0, any description="Allow the dSPACE Net Service to connect to a dSPACE expansion box via network."`
- `netsh advfirewall firewall add rule name="dSPACE MotionDesk" program=<main installation path>\dSPACERCPHIL2017-B\MotionDesk\Bin\MotionDesk.exe" dir=in action=allow profile=any description="Allow dSPACE MotionDesk to receive motion data via network."`

**Required open TCP/IP network ports** If you are using third-party firewall software on your host PC, ensure that the TCP/IP communication of dSPACE software is not blocked:

- VEOS requires the following open TCP/IP network ports: 111 (TCP and UDP), 3702 (UDP), 7214 (TCP and TCP6), 9923 (UDP), 15000 (UDP), 49152 ... 65535 (TCP, TCP6 and UDP)
- dSPACE Installation Manager and CodeMeter licensing software require the following open TCP/IP network port if communication in a LAN network is necessary: 22350 (TCP and UDP), if not changed from the default setting.

---

### Operating system on server for floating network licenses

If you purchased floating network licenses, you have to specify one of the network PCs as a license server. Every PC with CodeMeter Runtime software can be used as a license server.

**Valid for servers without dSPACE software** dSPACE only tests license servers with Microsoft Windows operating systems in combination with protected dSPACE software.

**Note**

Non-Windows operating systems, such as Ubuntu Linux, are not tested. You can use them at your own risk. dSPACE does not provide support in this case.

**Valid for servers with dSPACE Installation Manager** dSPACE Installation Manager supports the same operating systems as the other dSPACE software products as described above.

## Notes on 64-Bit Compatibility of dSPACE Products

**Notes**

As of dSPACE Release 2016-B, all products are available only as 64-bit variants. As a consequence, dSPACE Release 2016-B and later supports only the 64-bit variants of the following software:

- Windows operating systems
- MATLAB
- Python

This means:

- Product extensions, e.g., ConfigurationDesk custom function blocks, must be available in 64-bit versions.
- Python:
  - No support of 32-bit third-party extensions in the 64-bit Python installation from dSPACE.
  - No support of 64-bit dSPACE Python extensions in a parallel 32-bit Python installation.

## Run-Time Compatibility of dSPACE Software

**Definition**

Run-time compatibility means that:

- dSPACE products can be used in parallel after software installation, even if they are installed in different folders.
- dSPACE products without interaction can run independently of each other.

**Compatibility of products in dSPACE Release 2017-B**

dSPACE recommends using only software products from the same dSPACE Release. This ensures maximum run-time compatibility.

Observe the following points:

- Limitations regarding run-time compatibility in the dSPACE tool chain might occur if products from different dSPACE Releases are mixed.

If dSPACE products interact directly (through automation interfaces) or indirectly (through common file types like A2L), limitations might apply. For minor limitations, refer to the relevant product documentation. The major limitations are described in the following.

In rare cases, an additional patch must be installed for a product to achieve run-time compatibility. For more information on the patch and whether a patch is necessary, refer to <http://www.dspace.com/go/CompPatch>.

- RCP and HIL software products (of Release 2017-B) cannot be used in combination with RCP and HIL software products from earlier dSPACE Releases.

**Major limitation for working with a SCALEXIO system** The products for working with a SCALEXIO system must be compatible. This is guaranteed only for products delivered with the same dSPACE Release. Contact dSPACE for more information if you have any questions.

**Compatibility of real-time applications loaded to a DS1005, DS1006, DS1103, DS1104 or MicroAutoBox platform** If a real-time application is loaded to one of these platforms with a software product of dSPACE Release 2016-B or later, software products of dSPACE Release 2016-A (and earlier) do not detect that the loaded real-time application is the same as the real-time application stored on your host PC. In this case, you cannot work with the related software product without restrictions.

This also applies if you load a real-time application with a software product of dSPACE Release 2016-A or earlier and use software products of dSPACE Release 2016-B or later, for example, for experimenting.

---

**Combining dSPACE products from earlier Releases**

For more information and notes on the combined use of different products from and with earlier Releases, refer to [http://www.dspace.com/go/ds\\_sw\\_combi](http://www.dspace.com/go/ds_sw_combi).

## Limitations for Using Windows Features

---

**Motivation**

Some limitations apply using dSPACE software in conjunction with features of Windows.

**Fast user switching not supported**

dSPACE software does not support the fast user switching feature of Windows.

**Closing dSPACE software before PC shutdown**

The shutdown process of Windows operating systems might cause some required processes to be aborted although they are still being used by dSPACE

software. To avoid a loss of data, it is recommended to close the dSPACE software manually before shutting down the PC.

---

**User Account Control**

It is recommended to disable the Windows User Account Control (UAC) during the installation of dSPACE software. If you cannot disable UAC, note the following Windows behavior: If UAC is enabled, the setup programs use the administrator account instead of the user account. Therefore, it is important that the administrator account has access to the required drives, particularly the required network drives.

---

**USB devices**

If you connect dSPACE USB devices that use cables with optoisolation to the PC for the first time, there might be a message that the device driver software was not installed successfully. However, the dSPACE device will work properly later on.

---

**Long paths**

dSPACE software does not support the long path syntax of the Windows API. If a path that exceeds 260 characters is used directly or indirectly, the behavior of the dSPACE software is not defined.

---

**FIPS support**

dSPACE software was not developed for or tested against the FIPS PUB 140-2 U.S. government computer security standard (Security Requirements for Cryptographic Modules). Therefore, dSPACE products are not guaranteed to work if the respective setting is enabled in Windows. By default, the setting is disabled. For more information on FIPS, refer to <https://technet.microsoft.com/en-us/library/security/cc750357.aspx>.

---

**Enabling Windows 8dot3name creation option**

**Note**

It is strongly recommended that the Windows **8dot3name creation** option is enabled for all drives (drives used for installation and drives used for work) before you install third-party software, such as MATLAB®/Simulink®, and the dSPACE software.

If the option is disabled during software installation, serious errors can occur when you run the dSPACE software. For example, the build process might be aborted. To repair an installation that was installed while the **8dot3name creation** option was disabled, you have to install dSPACE software and required third-party software again.

For instructions on checking the setting and enabling the option, refer to <http://www.dspace.com/faq?346> or to the Microsoft Windows documentation.



**Settings in Windows for user locale and system locale must match**

MATLAB reads the user locale and system locale settings that are specified in Windows operating systems. The user locale and the system locale must match. If these settings are not the same, the system might not behave as expected when working with MATLAB and dSPACE software.

For instructions on checking and changing the settings, refer to [https://www.mathworks.com/help/matlab/matlab\\_env/setting-locale-on-windows-platforms.html?s\\_tid=gn\\_loc\\_drop](https://www.mathworks.com/help/matlab/matlab_env/setting-locale-on-windows-platforms.html?s_tid=gn_loc_drop).

This affects all MATLAB versions and all Windows operating systems, that are supported by dSPACE.



## Numerics

64-bit compatibility 270

## A

ASM blocksets  
 migration 54  
 new features 54  
 ASM Brake Hydraulics Blockset  
 migration 55  
 ASM Diesel Engine Blockset  
 migration 60  
 new features 56  
 ASM Diesel Exhaust Blockset  
 migration 62  
 new features 62  
 ASM Drivetrain Basic Blockset  
 migration 65  
 new features 64  
 ASM Engine Gasoline Blockset  
 migration 70  
 new features 68  
 ASM Environment Blockset  
 new features 66  
 ASM Traffic Blockset  
 migration 74  
 new features 73  
 ASM Trailer Blockset  
 migration 75  
 ASM Truck Blockset  
 migration 76  
 ASM Vehicle Dynamics Blockset  
 migration 77  
 new features 77  
 AutomationDesk  
 migration 51  
 new features 47  
 AUTOSAR  
 TargetLink-related  
 migration 226

## B

Bus Manager (stand-alone)  
 migration 81  
 new features 79

## C

CodeMeter Runtime  
 internal communication 42  
 Common Program Data folder 12  
 CommonProgramDataFolder 12  
 ConfigurationDesk - Configuration Version  
 new features 95  
 ControlDesk  
 migration 113  
 new features 98

## D

DCI Configuration Tool  
 new features 121  
 DCI-CAN1  
 discontinuation 17  
 DCI-GSI1  
 discontinuation 17  
 discontinuation  
 hardware 17  
 Documents folder 12  
 DocumentsFolder 12  
 dSPACE ECU Flash Programming Tool  
 new features 125  
 dSPACE FlexRay Configuration Package  
 new features 127  
 dSPACE License Central  
 Internet connection 42  
 dSPACE License Manager (Legacy) 42  
 dSPACE XIL API  
 new features 129

## E

ECU Interface Manager  
 migration 132  
 new features 131

## G

general enhancements and changes 15

## H

host PC software  
 MATLAB 267  
 operating system 268

## K

key features 23

## L

Limitations  
 TargetLink  
 obsolete limitations 256  
 limitations for using Windows features 271  
 Local Program Data folder 12  
 LocalProgramDataFolder 12

## M

MATLAB  
 requirements 267  
 startup 212  
 supported releases 267  
 migration  
 ASM blocksets 54  
 ASM Brake Hydraulics Blockset 55  
 ASM Diesel Engine Blockset 60  
 ASM Diesel Exhaust Blockset 62  
 ASM Drivetrain Basic Blockset 65  
 ASM Engine Gasoline Blockset 70

ASM Traffic Blockset 74  
 ASM Trailer Blockset 75  
 ASM Truck Blockset 76  
 ASM Vehicle Dynamics Blockset 77  
 AutomationDesk 51  
 Bus Manager (stand-alone) 81  
 ControlDesk 113  
 ECU Interface Manager 132  
 Model Compare 135  
 ModelDesk 139  
 MotionDesk 149  
 Real-Time Testing 152  
 RTI 156  
 RTI Bypass Blockset 157  
 RTI CAN MultiMessage Blockset 160  
 RTI FPGA Programming Blockset 165  
 RTI LIN MultiMessage Blockset 168

Model Compare  
 migration 135  
 new features 133  
 model management  
 new features 178

ModelDesk  
 migration 139  
 new features 137  
 MotionDesk  
 migration 149  
 new features 147

## N

new features  
 ASM blocksets 54  
 ASM Diesel Engine Blockset 56  
 ASM Diesel Exhaust Blockset 62  
 ASM Drivetrain Basic Blockset 64  
 ASM Engine Gasoline Blockset 68  
 ASM Environment Blockset 66  
 ASM Traffic Blockset 73  
 ASM Vehicle Dynamics Blockset 77  
 AutomationDesk 47  
 Bus Manager (stand-alone) 79  
 ConfigurationDesk - Configuration Version  
 95  
 ControlDesk 98  
 DCI Configuration Tool 121  
 dSPACE ECU Flash Programming Tool 125  
 dSPACE FlexRay Configuration Package 127  
 dSPACE XIL API 129  
 ECU Interface Manager 131  
 Model Compare 133  
 model management 178  
 ModelDesk 137  
 MotionDesk 147  
 Real-Time Testing 151  
 RTI Bypass Blockset 157  
 RTI CAN MultiMessage Blockset 159  
 RTI FPGA Programming Blockset 163  
 RTI LIN MultiMessage Blockset 167  
 RTI/RTI-MP 155  
 RTLib 155  
 SCALEXIO firmware 169

- signal & parameter management 178
- SystemDesk 186
- test management 176
- VEOS 261
- workflow management 181
- new general features
  - SYNECT 172
- not supported MATLAB features (R2017b)
  - RTI/RTI-MP 155

**P**

- product overview 21

**R**

- Real-Time Testing
  - migration 152
  - new features 151
- requirements
  - host PC software
    - MATLAB 267
    - operating system 268
- RTI Bypass Blockset
  - migration 157
  - new features 157
- RTI CAN MultiMessage Blockset
  - migration 160
  - new features 159
- RTI FPGA Programming Blockset
  - migration 165
  - new features 163
- RTI LIN MultiMessage Blockset
  - migration 168
  - new features 167
- RTI/RTI-MP
  - new features 155
  - not supported MATLAB features (R2017b) 155
- RTLib
  - new features 155

**S**

- SCALEXIO firmware
  - new features 169
- signal & parameter management
  - new features 178
- supported MATLAB releases 267
- SYNECT
  - new general features 172
- system requirements
  - operating system 268
- SystemDesk
  - new features 186

**T**

- TargetLink
  - API commands
    - changes 225
  - AUTOSAR features, new
    - supported releases 202

- code changes
  - migration 232
- code generator options
  - backward compatibility 222
  - changed default value 222
- discontinued features 255
- migrating to
  - new version 212
- migration
  - AUTOSAR-related 226
  - code changes 232
  - obsolete limitations 256
  - Property Manager 226
  - various aspects 228
- new API functions 207
- new code generator options 206
- new features 196
  - general changes 209
  - general enhancements 209
- new version
  - migrating to 212
- target support
  - discontinued compiler versions 205
  - discontinued evaluation boards 205
  - new compiler versions 205
  - new evaluation boards 205
  - supported targets 205
- TargetLink Data Dictionary
  - API commands
    - changes 225
  - migrating to
    - new version 212
  - migration 213
    - discontinued documentation 213
    - manually upgrading libraries and models 218
    - upgrading existing data dictionaries 216
  - new features 196
  - new version
    - migrating to 212
- test management
  - new features 176

**U**

- user documentation
  - printed documents 17

**V**

- VEOS
  - new features 261
- version history 21

**W**

- Windows
  - limitations 271
- workflow management
  - new features 181