

dSPACE Release

New Features and Migration

Release 2019-B – November 2019

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
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: <http://www.dspace.com/go/locations>
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form:

<http://www.dspace.com/go/supportrequest>. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the serial number of the hardware, the relevant dSPACE License ID, or the serial number of the CmContainer in your support request.

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/patches> for software updates and patches.

Important Notice

This publication contains proprietary information that is protected by copyright. All rights are reserved. The publication may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the publication 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 - 2019 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

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

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

About This Document	11
Conventions Used in dSPACE User Documentation.....	11
Accessing dSPACE Help and PDF Files.....	12
Overview of dSPACE Release 2019-B	15
General Enhancements and Changes.....	15
Discontinuations.....	17
Product Version Overview.....	18
New Product Key Features.....	20
Aspects of Migrating from Previous Releases	27
Migrating to dSPACE Release 2019-B.....	27
AutomationDesk	29
New Features of AutomationDesk 6.2.....	29
Migrating to AutomationDesk 6.2.....	31
Automotive Simulation Models (ASM)	33
ASM Base InCylinder.....	34
New Features of ASM Base InCylinder Blockset 2.6.....	34
Migrating to ASM Base InCylinder Blockset 2.6.....	34
ASM Brake Hydraulics.....	36
Migrating to ASM Brake Hydraulics Blockset 2.1.2.....	36
ASM Diesel Engine.....	37
New Features of ASM Diesel Engine Blockset 2.6.4.....	37
ASM Diesel Exhaust.....	38
New Features of ASM Diesel Exhaust Blockset 2.1.9.....	38
Changes in the ASM Diesel Exhaust Demo Model.....	38
Migrating to ASM Diesel Exhaust Blockset 2.1.9.....	38
ASM Diesel InCylinder.....	40
New Features of ASM Diesel InCylinder Blockset 2.7.1.....	40

ASM Drivetrain Basic.....	41
Migrating to ASM Drivetrain Basic Blockset 5.3.1.....	41
ASM Electric Components.....	42
New Features of ASM Electric Components Blockset 3.9.....	42
Changes in the ASM Electric Components Demo Model.....	42
Migrating to ASM Electric Components Blockset 3.9.....	43
ASM Environment.....	44
New Features of ASM Environment Blockset 4.11.....	44
Migrating to ASM Environment Blockset 4.11.....	44
ASM Gasoline Engine Basic.....	46
New Features of ASM Gasoline Engine Basic Blockset 2.2.4.....	46
ASM Gasoline Engine.....	47
New Features of ASM Gasoline Engine Blockset 4.0.4.....	47
ASM Gasoline InCylinder.....	48
New Features of ASM Gasoline InCylinder Blockset 2.7.1.....	48
ASM Traffic.....	49
New Features of ASM Traffic Blockset 4.0.....	49
Changes in the ASM Traffic Demo Model.....	50
Migrating to ASM Traffic Blockset 4.0.....	51
ASM Trailer.....	53
Changes in the ASM Trailer Demo Model.....	53
ASM Truck.....	54
Changes in the ASM Truck Demo Model.....	54
ASM Vehicle Dynamics.....	55
New Features of ASM Vehicle Dynamics Blockset 4.1.2.....	55
Changes in the ASM Vehicle Dynamics Demo Model.....	55
Bus Manager (Stand-Alone)	57
New Features of the Bus Manager (Stand-Alone) 6.4.....	57
Migrating to Bus Manager (Stand-Alone) 6.4.....	59
ConfigurationDesk	61
ConfigurationDesk - Implementation Version.....	62
New Features of ConfigurationDesk 6.4 (Implementation Version).....	62
Migrating to ConfigurationDesk 6.4.....	74

ControlDesk	77
New Features of ControlDesk 7.1.....	78
New Project and Experiment Features (ControlDesk 7.1).....	78
New Features of Platform Management and Platforms/Devices (ControlDesk 7.1).....	79
New Instrument Features (ControlDesk 7.1).....	79
New Measurement and Recording Features (ControlDesk 7.1).....	80
New Bus Navigator Features (ControlDesk 7.1).....	80
Additional Enhancements and Changes with ControlDesk (ControlDesk 7.1).....	83
Migrating to ControlDesk 7.1.....	84
Discontinuations in ControlDesk.....	84
Migrating to ControlDesk 7.1.....	86
dSPACE FlexRay Configuration Package	91
New Features of dSPACE FlexRay Configuration Package 4.4.....	91
dSPACE Python Extensions	93
New Features of dSPACE Python Extensions 3.2.....	93
dSPACE Installation Manager	95
New Features of dSPACE Installation Manager 5.4.....	95
Migrating to dSPACE Installation Manager 5.4.....	96
dSPACE XIL API .NET	99
New Features of dSPACE XIL API .NET 2019-B.....	99
ECU Interface Manager	101
New Features of ECU Interface Manager 2.6.....	101
Compatibility of ECU Interface Manager 2.6.....	102
Migrating to ECU Interface Manager 2.6.....	103
Firmware Manager	105
New Features of Firmware Manager 3.0.....	105

Model Compare	107
New Features of Model Compare 3.0.....	107
Migration to Model Compare 3.0.....	109
ModelDesk	111
New Features of ModelDesk 5.2.....	111
Migration to ModelDesk 5.2.....	112
Model Interface Package for Simulink	113
New Features of the Model Interface Package for Simulink 4.2.....	113
Migrating to the Model Interface Package for Simulink 4.2.....	114
MotionDesk	117
New Features of MotionDesk 4.5.....	117
Migrating to MotionDesk 4.5.....	117
Real-Time Testing	119
New Features of Real-Time Testing 4.2.....	119
Migrating to Real-Time Testing 4.2.....	119
RTI/RTI-MP and RTLib	121
New Features of RTI/RTI-MP and RTLib.....	121
Migration Aspects of RTI/RTI-MP and RTLib.....	122
RTI Bypass Blockset	125
New Features of the RTI Bypass Blockset 3.13.....	125
Migrating to RTI Bypass Blockset 3.13.....	125
RTI CAN MultiMessage Blockset	127
New Features of the RTI CAN MultiMessage Blockset 5.3.....	127
Migrating to RTI CAN MultiMessage Blockset 5.3.....	128

RTI FPGA Programming Blockset	129
New Features of the RTI FPGA Programming Blockset 3.8.....	129
Migrating to RTI FPGA Programming Blockset 3.8.....	131
RTI LIN MultiMessage Blockset	133
Migrating to RTI LIN MultiMessage Blockset 3.3.....	133
RTI Synchronized Time Base Manager Blockset	135
New Features of the RTI Synchronized Time Base Manager Blockset 1.3.....	135
SCALEXIO Firmware	137
New Features of the SCALEXIO Firmware 4.5.....	137
SYNECT	139
New Features of SYNECT 2.8.....	140
New General Features of SYNECT.....	140
New Features of Test Management.....	141
New Features of Workflow Management.....	143
Migrating to SYNECT 2.8.....	145
Migrating Databases.....	145
Migrating from SYNECT 2.7.....	146
Data Model Changes From SYNECT 2.7 to SYNECT 2.8.....	149
SystemDesk	151
New Features of SystemDesk 5.4.....	152
New General Features.....	152
Splittable AUTOSAR Elements.....	152
Configuring ECUs.....	154
Migrating to SystemDesk 5.4.....	156
Migrating to SystemDesk 5.4.....	156

TargetLink	157
New Features of TargetLink 5.0.....	158
Modeling in Simulink or Stateflow.....	158
Improved Support of Array-of-Structs.....	158
Data Stores Usable Across Code Generation Units.....	159
N-Dimensional Look-Up Tables.....	159
Other MATLAB®/Simulink/Stateflow Features.....	159
MATLAB® Code.....	160
MATLAB® Code Improvements.....	160
Adaptive AUTOSAR.....	161
Adaptive AUTOSAR.....	161
Classic AUTOSAR.....	162
Supported AUTOSAR Releases.....	162
Improved Support of Data Store Blocks for AUTOSAR Communication.....	163
Improved Import and Export for AUTOSAR 4.4.....	163
Tool Chain Support.....	164
Using Several TargetLink-Generated SIC Files in the Same Application Process.....	164
Code Generation Core Functionality.....	164
Support of Modern Standards of C.....	164
Void casts in function calls.....	165
Generating Code for 64-Bit Linux Targets.....	165
Improved Code Efficiency.....	165
Target Simulation (PIL).....	166
Changes in the Target Simulation Modules.....	166
Usability.....	167
Data Dictionary and Data Dictionary Manager.....	168
Property Manager.....	169
Code Generator Options.....	170
New Code Generator Options.....	170
API Functions and Hook Scripts.....	171
New API Functions.....	171
New Hook Scripts.....	172
Improved API Functions and Hook Scripts.....	172

Other.....	173
General Enhancements and Changes.....	173
TargetLink Demos.....	174
Migrating to TargetLink 5.0 and TargetLink Data Dictionary 5.0.....	175
General Migration Information.....	175
Upgrading Models, Libraries, and Data Dictionaries.....	175
Basics on Migrating Between TargetLink Versions.....	175
How to Upgrade a Data Dictionary with Included DD Files.....	178
How to Manually Upgrade Libraries and Models via the API.....	179
Migrating Data Dictionaries to CodeDecorationSets.....	180
Migrating from TargetLink 4.4 to 5.0.....	183
Code Generator Options.....	184
Migration Aspects Regarding Code Generator Options.....	184
AUTOSAR.....	187
AUTOSAR 2.x/3.x no Longer Supported.....	187
Changed Block Property for AUTOSAR Mode.....	188
Custom Code.....	188
Dereferencing of Pointers.....	188
API Functions and Hook Scripts.....	189
Changes in API Functions.....	189
Changes in Hook Scripts.....	190
Messages.....	190
Message changes.....	190
Other Migration Aspects.....	191
Various Migration Aspects.....	191
Code Changes Between TargetLink 4.4 and TargetLink 5.0.....	193
Shift Operations.....	194
Extern C Encapsulation.....	196
Access Functions.....	196
AUTOSAR.....	197
Global Bitfields.....	198
MATLAB Code.....	199
Name Template of DD VariableTemplate Objects.....	199
Look-Up Tables.....	200
Variable Vector Width.....	201
Stateflow.....	201
Other.....	204

Discontinuations as of TargetLink 5.0.....	207
Discontinued TargetLink Features	208
Obsolete API Functions.....	209
Obsolete Limitations.....	210
Changes in Future TargetLink Versions.....	211
Features to Be Discontinued.....	211
API Functions to Be Discontinued.....	212
Deprecated Code Generator Options.....	212
VEOS	213
New Features of VEOS 4.5.....	213
Compatibility of VEOS 4.5.....	214
Migrating to VEOS 4.5.....	217
Discontinuations in VEOS 4.5.....	218
Compatibility Information	219
Supported MATLAB Releases.....	219
Operating System.....	220
Using dSPACE Software on Virtual Machines (VMs).....	223
Run-Time Compatibility of dSPACE Software.....	226
Limitations for Using Windows Features.....	227
Index	231




About This Document






Contents	This document informs you about the new features of all the dSPACE software products in Release 2019-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.
Printed document	A printed copy of this document is available on demand. You can order it free of charge by using the following link: http://www.dspace.com/go/requestreleasematerial .
Where to go from here	Information in this section <ul style="list-style-type: none"> 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
	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.

Symbol	Description
	Indicates a hazard that, if not avoided, could result in property damage.
	Indicates important information that you should take into account to avoid malfunctions.
	Indicates tips that can make your work easier.
	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

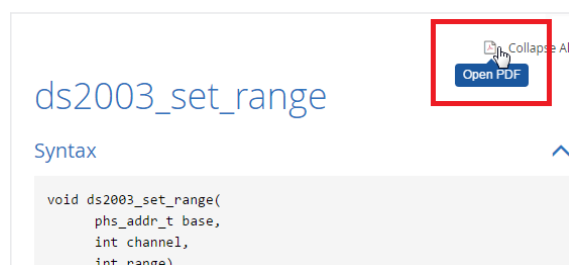
Opening context-sensitive help dSPACE Help provides context-sensitive help. You can open help on the active context in dSPACE software:

- 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 that contains the user documentation in PDF format via the  button in the Backstage view (leftmost ribbon tab).

Overview of dSPACE Release 2019-B

Introduction Gives you an overview of the new key features in Release 2019-B and information about unchanged products.

Where to go from here

Information in this section

General Enhancements and Changes	15
Discontinuations	17
Product Version Overview	18
New Product Key Features	20

General Enhancements and Changes

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

Support of new dSPACE hardware

dSPACE Release 2019-B introduces new dSPACE hardware:

- **MicroAutoBox III**
The new MicroAutoBox III is the successor of the MicroAutoBox II, a real-time system for performing fast, in-vehicle function prototyping:
 - The new DS1403 Processor Board with the TI AM5K2E04 processor and four ARM® Cortex®-A15 processor cores functions as the real-time processor. The DS1403 provides two standard Ethernet I/O interfaces and two automotive Ethernet interfaces.
 - Support of the DS1511, DS1513, and DS1514 I/O Boards known from the MicroAutoBox II.
 - Supported by ConfigurationDesk to implement real-time applications.

For an overview of the supported I/O functionality, refer to [Overview of MicroAutoBox III I/O Functionality](#) ([📖 MicroAutoBox III Hardware Installation and Configuration](#)).

For information for former MicroAutoBox II users, refer to [Main Differences Between Modeling with RTI and ConfigurationDesk](#) ([📖 MicroAutoBox III - Getting Started](#)).

Python distribution

The libraries and components used with Python 3.6 and distributed on dSPACE DVDs have changed as shown in the following table.

Package	Release 2019-A	Release 2019-B
comtypes	1.1.7	1.1.7
Core	3.6.7	3.6.8
cycler	0.10.0	0.10.0
kiwisolver	1.0.1	1.1.0
matplotlib	3.0.1	3.1.0
numpy	1.15.4	1.16.4
pillow	5.3.0	6.0.0
pip	18.1	19.1.1
pyparsing	2.3.0	2.4.0
pypubsub	4.0.0	4.0.3
Python-dateutil	2.7.5	2.8.0
pythonnet	2.3.2	2.4.1
pytz	2018.7	2019.1
pywin32	224.10	224.10
six	1.11.0	1.12.0
wxPython	4.0.3	4.0.6

Using dSPACE software on virtual machines (VM)

As of dSPACE Release 2019-A, you can operate several dSPACE products on virtual machines. For more information, refer to [Using dSPACE Software on Virtual Machines \(VMs\)](#) on page 223.

Legacy licensing using CodeMeter licenses simplifies installation and use of dSPACE releases earlier than 2017-B

If you own a software product from dSPACE Release 2017-B or later and you want to install and use a version of this product from a dSPACE Release earlier than 2017-B, you must have legacy license files and a CodeMeter CmContainer with activated licenses. Until now, you had to contact dSPACE and provide specific information before dSPACE sent you the license files.

As of October 2019, you can use dSPACE Installation Manager 5.4 to download license files specifically prepared on the basis of the licenses you purchased. Legacy licensing using CodeMeter licenses maps former product versions to an available license so that you can install and use products from dSPACE Release 7.4 (2012-B) up to dSPACE Release 2017-A.

Refer to [Legacy Licensing Using CodeMeter Licenses](#) ( Working with CodeMeter Licensing Technology).

RCP and HIL software: C/C++ compilers for building MATLAB MEX files

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

- MinGW (GNU Compiler Collection (GCC 5.3.0)): MATLAB R2018a.
- MinGW (GNU Compiler Collection (GCC 6.3.0)): MATLAB R2018b, R2019a, and R2019b.
- Microsoft Visual Studio 2015 Professional.

Discontinuations

Introduction

The following discontinuations for software and hardware are relevant to the current Release or are planned for future Releases.

For more end-of-life announcements, refer to <http://www.dspace.com/go/discontinuation>.

Planned discontinuation of dSPACE hardware

As of Release 2020-A the software support will be discontinued for the following products, which were discontinued in previous Releases:

- DS1005 PPC Board
For new projects, we recommend that you use the dSPACE DS1007 PPC Processor Board.
- MicroAutoBox II variants with the DS1512 I/O Board
For new projects, we recommend that you use the MicroAutoBox II or MicroAutoBox III variants with DS1514 I/O Board.
- DCI-CAN1
For new projects, we recommend that you use the DCI-CAN2 or DCI-CAN/LIN1.
- DCI-GSI1
For new projects, we recommend that you use the DCI-GSI2.

Planned software support discontinuation

Discontinuation of Windows 7 Because the official Microsoft support for Windows 7 ends in January 2020, dSPACE will discontinue the support for Windows 7 SP1 as operating system as of dSPACE Release 2020-A.

Discontinuation of .NET 2.0 APIs With dSPACE Release 2020-A, dSPACE will discontinue the support for client programs and libraries built with .NET Runtime 2.0. This applies to any C#-based programs using the provided interfaces (APIs) for COM automation and for the Installation Manager. Applications using these interfaces have to use at least .NET Runtime 4.0.

SCALEXIO firmware operating system change in Release 2020-B To speed-up the development of new features and to provide an improved real-time execution for SCALEXIO, dSPACE changes the basic operating system of the SCALEXIO firmware in Release 2020-B to Linux™.

This will affect real-time applications, model containers (i.e. BSC, FMU, SIC, V-ECU), s-functions and custom I/O-functions that contain binary libraries built for dSPACE Release 2020-A and older. All real-time applications, precompiled model containers and binary libraries used for SCALEXIO are required to be (re)-built from source code based on dSPACE Release 2020-B. More information for supporting the migration will be provided with Release 2020-A and 2020-B.

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			
	2018-A	2018-B	2019-A	2019-B
AutomationDesk	5.6	6.0	6.1	6.2 Refer to AutomationDesk on page 29.
Automotive Simulation Models	9.1	9.2	9.3	9.4 Refer to Automotive Simulation Models (ASM) on page 33.
Bus Manager (stand-alone)	6.1	6.2	6.3	6.4 Refer to Bus Manager (Stand-Alone) on page 57.
ConfigurationDesk	6.1	6.2	6.3	6.4 Refer to ConfigurationDesk on page 61.
Container Manager	5.0	5.1	5.1	5.1
ControlDesk	6.3	6.4	7.0	7.1 Refer to ControlDesk on page 77.
DCI Configuration Tool	3.9	3.10	3.11	3.11
dSPACE CAN API Package	3.0.3	4.0.1	4.0.2	4.0.3
dSPACE ECU Flash Programming Tool	2.5	2.5	2.6	2.6
dSPACE FlexRay Configuration Package	4.1	4.2	4.3	4.4 Refer to dSPACE FlexRay Configuration Package on page 91.
dSPACE Python Extensions	2.5	3.0	3.1	3.2 Refer to dSPACE Python Extensions on page 93.

Product	dSPACE Release			
	2018-A	2018-B	2019-A	2019-B
dSPACE XIL API .NET	2018-A	2018-B	2019-A	2019-B Refer to dSPACE XIL API .NET on page 99.
ECU Interface Manager	2.3	2.4	2.5	2.6 Refer to ECU Interface Manager on page 101.
Firmware Manager	2.5	2.6	2.7	3.0 Refer to Firmware Manager on page 105.
Model Compare	2.8	2.9	2.9	3.0 Refer to Model Compare on page 107.
ModelDesk	4.7	5.0	5.1	5.2 Refer to ModelDesk on page 111.
Model Interface Package for Simulink	3.6	4.0	4.1	4.2 Refer to Model Interface Package for Simulink on page 113.
MotionDesk	4.2	4.3	4.4	4.5 Refer to MotionDesk on page 117.
MotionDesk Blockset	2.5.3	2.5.4	2.5.5	2.6 Refer to MotionDesk on page 117.
Real-Time Testing	3.4	4.0	4.1	4.2 Refer to Real-Time Testing on page 119.
RTI ¹⁾	7.10	7.11	7.12	7.13 Refer to RTI/RTI-MP and RTLib on page 121.
RTI-MP ²⁾	7.10	7.11	7.12	7.13 Refer to RTI/RTI-MP and RTLib on page 121.
RTI Bypass Blockset	3.10	3.11	3.12	3.13 Refer to RTI Bypass Blockset on page 125.
RTI CAN Blockset	3.4.6	3.4.7	3.4.8	3.4.9
RTI CAN MultiMessage Blockset	5.0	5.1	5.2	5.3 Refer to RTI CAN MultiMessage Blockset on page 127.
RTI Electric Motor Control Blockset	1.4.1	1.4.1	1.4.1	1.4.1
RTI Ethernet Blockset	1.2.3	1.2.3	1.2.3	1.2.3
RTI Ethernet (UDP) Blockset	1.4.3	1.4.3	1.4.3	1.4.3
RTI FPGA Programming Blockset	3.5	3.6	3.7	3.8 Refer to RTI FPGA Programming Blockset on page 129.
RTI LIN MultiMessage Blockset	3.0	3.1	3.2	3.3 Refer to RTI LIN MultiMessage Blockset on page 133.
RTI RapidPro Control Unit Blockset	2.2.3	2.2.3	2.2.3	2.2.3

Product	dSPACE Release			
	2018-A	2018-B	2019-A	2019-B
RTI Synchronized Time Base Manager Blockset	-	1.1	1.2	1.3 Refer to RTI Synchronized Time Base Manager Blockset on page 135.
RTI USB Flight Recorder Blockset	1.2.2	1.2.2	1.2.2	1.2.2
RTI Watchdog Blockset	2.1.1	2.1.1	2.1.1	2.1.1
Sensor Simulation	-	1.0	1.1	1.2
SCALEXIO firmware	4.2	4.3	4.4	4.5 Refer to SCALEXIO Firmware on page 137.
SYNECT	2.5	2.6	2.7	2.8 Refer to SYNECT on page 139.
SystemDesk	5.1	5.2	5.3	5.4 Refer to SystemDesk on page 151.
TargetLink	4.3	4.4	4.4	5.0 Refer to TargetLink on page 157.
Variable Editor ³⁾	2.4	2.4	2.4	2.4
VEOS	4.2	4.3	4.4	4.5 Refer to VEOS on page 213.

¹⁾ Including the standard I/O blocksets.

²⁾ Including the RTI Gigalink Blockset.

³⁾ The Variable Editor is not 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 the new key features for each product. For more information, refer to the product-specific sections.

Where to go from here

Information in this topic

AutomationDesk	21
Bus Manager (stand-alone)	21
ConfigurationDesk (Implementation Version)	21
ControlDesk	22
dSPACE FlexRay Configuration Package	22
dSPACE XIL API	23
ECU Interface Manager	23

Firmware Manager	23
Model Compare	23
Model Interface Package for Simulink	23
ModelDesk	23
MotionDesk	24
Python Extensions	24
Real-Time Testing	24
RTI, RTI-MP, and RTLib	24
RTI Bypass Blockset	24
RTI CAN MultiMessage Blockset	24
RTI FPGA Programming Blockset	25
RTI Synchronized Time Base Manager Blockset	25
SCALEXIO firmware	25
SYNECT	25
SystemDesk	25
TargetLink	25
VEOS	26

AutomationDesk

The new key feature of AutomationDesk is:

- The new XML file format introduced with Release 2019-A for exporting and importing AutomationDesk elements is now also used as the standard serialization format.

For more information on the new feature, refer to [New Features of AutomationDesk 6.2](#) on page 29.

Bus Manager (stand-alone)

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

- Support of J1939
- New and enhanced bus configuration features

For more information, refer to [New Features of the Bus Manager \(Stand-Alone\) 6.4](#) on page 57.

**ConfigurationDesk
(Implementation Version)**

The new key features of ConfigurationDesk are:

- Support of MicroAutoBox III.
- Various new function blocks and enhanced function blocks to support MicroAutoBox III.
- Support of BSC files in multimodel application processes.
- Support of TargetLink-generated SIC files in multimodel application processes.
- Various enhancements of the Bus Manager for configuring bus communication for simulation, inspection, and manipulation purposes.

For more information, refer to [ConfigurationDesk - Implementation Version](#) on page 62.

ControlDesk

The new key features of ControlDesk 7.1 are:

General enhancements

- Support of MicroAutoBox III

Project/experiment management enhancements

- Saving a project in another project root directory

For more information, refer to [New Project and Experiment Features \(ControlDesk 7.1\)](#) on page 78.

Platform/device enhancements

- New platform to support MicroAutoBox III
- SCALEXIO: Support of DS6121 Multi-I/O Board and DS6321 UART Board
- SCALEXIO: Refreshing the display of client processes
- Ethernet Bus Monitoring device: Support for signals of TCP Ethernet PDUs

For more information, refer to [New Features of Platform Management and Platforms/Devices \(ControlDesk 7.1\)](#) on page 79.

Instrument enhancements

- New 3-D Viewer

For more information, refer to [New Instrument Features \(ControlDesk 7.1\)](#) on page 79.

Measurement and recording enhancements

- Importing GPX files

For more information, refer to [New Measurement and Recording Features \(ControlDesk 7.1\)](#) on page 80.

Bus Navigator enhancements

- Reactivating FlexRay Bus Instrument support
- Support of new and enhanced bus configuration features in Bus Manager applications
- Ethernet Bus Monitoring device: Display of Ethernet PDUs and signals
- LIN bus statistics

For more information, refer to [New Bus Navigator Features \(ControlDesk 7.1\)](#) on page 80.

Additional enhancements

- New tutorial videos for the Internal Interpreter

For more information, refer to [Additional Enhancements and Changes with ControlDesk \(ControlDesk 7.1\)](#) on page 83.

dSPACE FlexRay Configuration Package

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

- Support of MicroAutoBox III.

For more information on the new feature, refer to [New Features of dSPACE FlexRay Configuration Package 4.4](#) on page 91.

dSPACE XIL API	<p>dSPACE XIL API MAPort supports MicroAutoBox III as the new dSPACE platform.</p> <p>For more information on the new features, refer to New Features of dSPACE XIL API .NET 2019-B on page 99.</p>
ECU Interface Manager	<p>The new key feature of the ECU Interface Manager is:</p> <ul style="list-style-type: none">▪ Support of MicroAutoBox III <p>For more information on the new feature, refer to New Features of ECU Interface Manager 2.6 on page 101.</p>
Firmware Manager	<p>The new key features of the Firmware Manager are</p> <ul style="list-style-type: none">▪ Support of new SCALEXIO boards.▪ Support of MicroAutoBox III. <p>For more information on the new features, refer to New Features of Firmware Manager 3.0 on page 105.</p>
Model Compare	<p>The new key features of Model Compare are:</p> <ul style="list-style-type: none">▪ Using separate MATLAB instances in parallel for the XML dump file creation to enhance performance significantly.▪ Using an external text comparison tool to make it markedly easier to work with properties of more than three lines, such as MATLAB scripts.▪ Starting the creation of XML dump files directly from the Simulink/Stateflow Editor to simplify when you are working with a version control system.▪ Specifying default positions for the relevant windows to help you keep your desktop well-organized when you work with multiple windows on multiple monitors.▪ The improved status bar displays additional useful information and lets you navigate to relevant configuration dialogs.▪ The new Start page facilitates starting work with Model Compare. <p>For more information on the new features, refer to New Features of Model Compare 3.0 on page 107.</p>
Model Interface Package for Simulink	<p>The new key feature of Model Interface Package for Simulink is:</p> <ul style="list-style-type: none">▪ Support of the row major format in Simulink <p>For more information on the new feature, refer to Model Interface Package for Simulink on page 113.</p>
ModelDesk	<p>The new key features of ModelDesk are:</p> <ul style="list-style-type: none">▪ Support of MicroAutoBox III▪ Parameterization: Parameterizing the new ASM traffic driver.

For more information on the new features, refer to [New Features of ModelDesk 5.2](#) on page 111.

MotionDesk

The new key features of MotionDesk are:

- Support of MicroAutoBox III

For more information on the new features, refer to [New Features of MotionDesk 4.5](#) on page 117.

Python Extensions

Python Extensions 3.2 does not have new features.

As of dSPACE Release 2018-B, Python Extensions supports Python 3.6. You have to migrate your custom scripts manually. For more information, refer to <http://www.dspace.com/go/Python36Migration> on the dSPACE website.

Real-Time Testing

The new key features of Real-Time Testing are:

- Support of MicroAutoBox III

For more information on the new features, refer to [New Features of Real-Time Testing 4.2](#) on page 119.

RTI, RTI-MP, and RTLib

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

- Support of MATLAB R2019b

For more information on the new feature, refer to [New Features of RTI/RTI-MP and RTLib](#) on page 121.

RTI Bypass Blockset

The new key feature of the RTI Bypass Blockset is:

- Support of ARM microcontrollers for on-target prototyping.

For more information on the new feature, refer to [New Features of the RTI Bypass Blockset 3.13](#) on page 125.

RTI CAN MultiMessage Blockset

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

- Support of static container IPDUs.
- Support of cryptographic IPDUs.
- Support of secure onboard communication for static container IPDUs.

For more information on the new features, refer to [New Features of the RTI CAN MultiMessage Blockset 5.3](#) on page 127.

RTI FPGA Programming Blockset

The new key features of the RTI FPGA Programming Blockset 3.8 are:

- Extended Xilinx® software support.
- New MicroAutoBox III frameworks.
- Enhancements to the SCALEXIO frameworks.

For more information on the new features, refer to [New Features of the RTI FPGA Programming Blockset 3.8](#) on page 129.

RTI Synchronized Time Base Manager Blockset

The new key feature of the RTI Synchronized Time Base Manager Blockset is:

- Support of MicroAutoBox III

For more information on the new feature, refer to [New Features of the RTI Synchronized Time Base Manager Blockset 1.3](#) on page 135.

SCALEXIO firmware

The new key feature of the SCALEXIO firmware is:

- Support of the DS6212 Multi-I/O Board
- Support of the DS6321 UART Board

For more information on the new features, refer to [New Features of the SCALEXIO Firmware 4.5](#) on page 137.

SYNECT

The new key features of SYNECT 2.8 are:

- Improved support of evaluations.
- Using workflows recursively as workflow steps.

For more information on the new features, refer to [New Features of SYNECT 2.8](#) on page 140.

SystemDesk

The new key features of SystemDesk 5.4 are:

- Support of the AUTOSAR AP 19-03 revision for developing Adaptive Platform software.
- Support of the splittable mechanism by assigning splittable AUTOSAR elements across several master files.
- Improved support for the generation of V-ECUs with additional dSPACE-specific MCAL BSW modules that can be used to simulate the generation of hardware events.

For more information on the new features, refer to [New Features of SystemDesk 5.4](#) on page 152.

TargetLink

The new key features of TargetLink 5.0 are:

- Support of Adaptive AUTOSAR
 - Modeling and generating code for service-based communication according to ara::com.

- Import and Export of Adaptive AUTOSAR ARXML files to and from the Data Dictionary.
- Data Store Memory blocks in the Data Dictionary
 - Creating and specifying DD Data Store Memory Block objects.
 - Using Data Store Memory blocks across CGUs via DD Data Store Memory Block objects.
- Arrays of struct and Data Store Memory blocks
 - Referencing array-of-structs variables in Data Store Memory blocks.
 - Element-wise access to array-of-structs variables via Data Store Read and Data Store Write blocks.
- Struct support for MATLAB® Code
 - Structures for Stateflow® data and MATLAB variables in MATLAB Code.
- Usability improvements for Data Dictionary dialogs
 - DD import and export dialogs
 - Advanced Edit String List dialog
- Validation Summary in the Property Manager
 - Display model element data validation errors and read-write errors from all model element variables with the option to filter, search and group errors.

For more information on all the new features, refer to [New Features of TargetLink 5.0](#) on page 158.

For more information on the TargetLink migration aspects (TargetLink, TargetLink AUTOSAR Module, TargetLink Data Dictionary), refer to [Migrating to TargetLink 5.0 and TargetLink Data Dictionary 5.0](#) on page 175.

VEOS

The new key features of VEOS are:

- Support of the Microsoft Visual C/C++ Compiler from Microsoft Visual Studio 2017
- Enhancements for adaptive V-ECUs
- Enhancements for classic V-ECUs

For more information on the new features, refer to [New Features of VEOS 4.5](#) on page 213.

Aspects of Migrating from Previous Releases

Introduction

After you install products of the current dSPACE Release, some additional steps might be required. 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 2019-B

Introduction

After you install Release 2019-B, some additional steps might be required.

Migrating from dSPACE Release 2019-A

Product-specific migration steps Product-specific migration steps are generally performed automatically. For exceptions, refer to the product-specific migration descriptions.

Migrating from dSPACE Release 2018-B or earlier

To migrate from dSPACE Release 2018-B or earlier to Release 2019-B, you also have to perform the migration steps of the intervening dSPACE Releases. All of the required migration steps can be performed using the software from dSPACE Release 2019-B.

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 2019-B\Print\PreviousReleases`.
- On the dSPACE DVDs. Refer to `\Doc\PreviousReleases`.
- At 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 6.2	29
Migrating to AutomationDesk 6.2	31

New Features of AutomationDesk 6.2

Where to go from here

Information in this topic

General enhancements	29
New XML format for serializing AutomationDesk projects and custom libraries	29
New platform support	30
Enhancements to the COM API	30

General enhancements

New XML format for serializing AutomationDesk projects and custom libraries

With AutomationDesk 6.1, a new XML format is introduced for exporting and importing AutomationDesk elements. This XML format is now also used as the standard serialization format for projects and custom libraries.

It is more flexible than the proprietary binary format and makes it easier to use tools for versioning, comparing, generating, or analyzing elements.

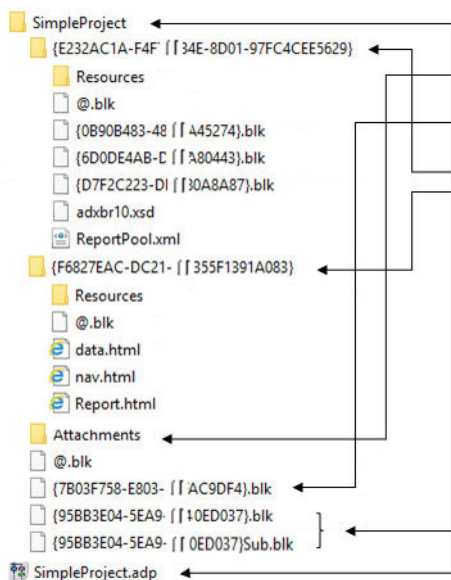
Some of the advantages are:

- Human-readable contents based on UTF-8 character encoding, such as date and time.
- The relationship between an XML file and the contained element can be identified by its name.
- Separation of structure information, user configuration, and sequences.

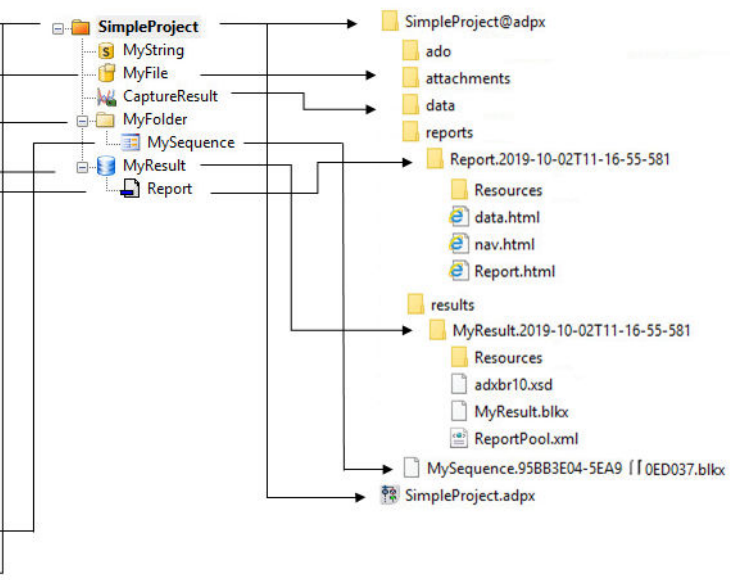
- Each sequence is stored in a single file.
- Data object values are contained in the related project or sequence XML files, except for Mapping and CaptureResult data objects, which are stored in separate files in the *data* folder.
- Reduced amount of elements within the XML files, e.g., no entries for elements that use the default values because the values are not explicitly specified.
- More efficient use of version control systems, such as, storing only the modified contents in the repository, merging different contents, etc.

The figure shows a simple AutomationDesk project and its related files in the legacy binary format on the left side and the new XML format on the right side.

AutomationDesk 6.1



AutomationDesk 6.2



The XML format used for exporting and importing elements with AutomationDesk 6.0 or earlier is called *legacy XML*. It is available only for importing existing XML export files. The legacy XML format is not available for exporting elements and will be discontinued in future versions of AutomationDesk.

For more information, refer to [Basics on the Serialization of AutomationDesk Projects](#) ([AutomationDesk Basic Practices](#)).

New platform support AutomationDesk supports MicroAutoBox III as the new dSPACE platform.

Enhancements to the COM API

The AutomationDesk COM API provides the following enhancements:

- You can use the `Load` and `LoadEx` methods to open AutomationDesk projects and custom libraries in the new XML format.

If you open a file in the legacy XML format, a warning is written to the log file, which informs you about the planned discontinuation.

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

Migrating to AutomationDesk 6.2

General migration aspects

If you open an AutomationDesk project with a later AutomationDesk version, the software automatically detects whether a migration is required. 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 properly.

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

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


Note

As of dSPACE Release 2018-B, AutomationDesk has supported Python 3.6. You find information on changes and migration aspects of Python scripts in dSPACE products on the dSPACE website. Refer to <http://www.dspace.com/go/Python36Migration>.

Migration to the new serialization format

If you open an AutomationDesk project or a custom library created with AutomationDesk 6.1 or earlier, the data is automatically migrated to the new XML format. The migration considers the standard serialization format, exported projects and custom libraries in ZIP format, and exported projects and custom libraries in the legacy XML format.

If you use a version control system, you can get and open projects and custom libraries in the legacy formats. They are automatically migrated to the new

serialization format and loose their connections to the version control system. You must add the migrated projects and custom libraries to new version control projects to avoid mixing the formats in the repository. For more information, refer to [How to Migrate Projects or Custom Libraries Under Version Control](#) ( [AutomationDesk Basic Practices](#)).

Libraries

ControlDesk Access library With ControlDesk 7.1, the following features changed. This also affects the ControlDesk Access library in AutomationDesk.

- Discontinuation of taking snapshots

As of ControlDesk 7.1, you can no longer take snapshots, i.e., read the current values of variable values defined in a label list, and save them to a CSV file. If you use the Snapshot block in AutomationDesk, an exception occurs. Use the ReadVariableValue block instead.

For more information, refer to  [AutomationDesk Accessing ControlDesk](#).

MATLAB Access library If you want to use the MATLAB Access library with MATLAB R2019a, you need MATLAB R2019a Update 5 or later. For more information, refer to [Supported MATLAB Releases](#) on page 219.

Planned discontinuations

As of Release 2020-A the following libraries will be discontinued:

- CANScope
- CANstress

Both libraries are not included in the standard distribution, but are available on demand. The AutomationDesk version as of Release 2020-A will no longer support these libraries.

Automotive Simulation Models (ASM)

Where to go from here

Information in this section

ASM Base InCylinder	34
ASM Brake Hydraulics	36
ASM Diesel Engine	37
ASM Diesel Exhaust	38
ASM Diesel InCylinder	40
ASM Drivetrain Basic	41
ASM Electric Components	42
ASM Environment	44
ASM Gasoline Engine Basic	46
ASM Gasoline Engine	47
ASM Gasoline InCylinder	48
ASM Traffic	49
ASM Trailer	53
ASM Truck	54
ASM Vehicle Dynamics	55

ASM Base InCylinder

Where to go from here

Information in this section

New Features of ASM Base InCylinder Blockset 2.6	34
Migrating to ASM Base InCylinder Blockset 2.6	34

New Features of ASM Base InCylinder Blockset 2.6

Crank-based high-pressure pump systems

The ASM InCylinder Base library contains two new blocks: HPP_CRANKBASED and CRANKBASED_RAIL_CONTROL

These blocks are used to set up engines with crank-based high-pressure pump systems.

ASM Engine Testbench

ASM Engine Testbench has new functionalities:

- A list for defining parameters and values has been added to the Online Settings page. These parameters are set to the specified values when the online test bench simulation is started on a dSPACE platform.
- The Plot Selected button has been added to the Execute page to only show the selected signals.

Migrating to ASM Base InCylinder Blockset 2.6

APU_EVENT block

The block has been extended with a switch for continuation of events in the next window to support crank-based high-pressure fuel pump systems.


RAIL block

The block has a new output: T_Rail[degC].

ENGINE_SETUP block

The block has been extended with the new `Const_max_num_HPP` parameter to support crank-based high-pressure fuel pump systems.

Related topics**Basics**

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

ASM Brake Hydraulics

Migrating to ASM Brake Hydraulics Blockset 2.1.2

Internal adaptations

Some blocks have had internal adaptations. These have no effect on the simulation result and its functionality.

This applies to the following blocks:

- PRE_CHARGE_VALVE_1
- PRE_CHARGE_VALVE_2
- NON_RETURN_VALVE_RES_1
- NON_RETURN_VALVE_RES_2
- PUMP_CONTROLLER
- TORQUE_REQUEST_ESP

Related topics

Basics

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

ASM Diesel Engine

New Features of ASM Diesel Engine Blockset 2.6.4

HPP_CRANKBASED block

The HPP_CRANKBASED block can now handle negative values of the phi_FMU_Energized control signal.

ASM Engine Testbench

ASM Engine Testbench has new functionalities:

- A list for defining parameters and values has been added to the Online Settings page. These parameters are set to the specified values when the online test bench simulation is started on a dSPACE platform.
- The Plot Selected button has been added to the Execute page to only show the selected signals.

ASM Diesel Exhaust

Where to go from here

Information in this section

New Features of ASM Diesel Exhaust Blockset 2.1.9	38
Changes in the ASM Diesel Exhaust Demo Model	38
Migrating to ASM Diesel Exhaust Blockset 2.1.9	38

New Features of ASM Diesel Exhaust Blockset 2.1.9

ELECTRIC_HEATER block

This is a new block. The ELECTRIC HEATER block calculates the heating power of the heating element. One application of the component functions as the heating of the diesel oxidation catalyst during the cold start phase.

Changes in the ASM Diesel Exhaust Demo Model

ExhaustSystem_DOC_DPF demo model

The demo model has a new block: ELECTRIC_HEATER. The block is placed in front of the DIESEL_OXIDATION_CATALYST block and is used as an external heat source of DOC.

ExhaustSystem_DOC_DPF_SCR demo model

The demo model has a new block: ELECTRIC_HEATER. The block is placed in front of the UREA_DECOMPOSITION block and used as an external heat source during the urea decomposition process.

ExhaustSystem_DOC_DPF_NonAir_SCR demo model

The demo model has a new block: ELECTRIC_HEATER. The block is placed in front of the UREA_DECOMPOSITION block and used as an external heat source during the urea decomposition process.

Migrating to ASM Diesel Exhaust Blockset 2.1.9

UREA_DECOMPOSITION block

There is a new Qdot_Heater[W] inport. With this inport, an external heat source, e.g., of an electric heater, can be added into the component.

**DIESEL_OXIDATION_
CATALYST block**

There is a new Qdot_Heater[W] inport. With this inport, an external heat source, e.g., of an electric heater, can be added into the component.

PUMP_HOSE block

The PUMP_HOSE/Mass Balance/Damping_System/Volume_Damper/Reset system has been revised. The Detect Increase Simulink block has been reproduced by other Simulink blocks. Additionally, you can trigger the reset from outside the block via the Reset[0|1] inport.

Related topics**Basics**

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

ASM Diesel InCylinder

New Features of ASM Diesel InCylinder Blockset 2.7.1

Crank-based high-pressure pump systems

The ASM InCylinder Diesel model now supports crank-based high-pressure pump systems. With the related blocks of the ASM InCylinder Base library, the ASM Diesel InCylinder demo systems are prepared. You can easily insert the subsystem into an ASM InCylinder Diesel demo model.

ASM Drivetrain Basic

Migrating to ASM Drivetrain Basic Blockset 5.3.1

Ambient block

The AMBIENT block has been split in two blocks: VEHICLE_POSITION and AMBIENT.

The VEHICLE_POSITION block calculates the altitude of the vehicle and the driven distance.

The AMBIENT block calculates the temperature and pressure of the ambient area.

In previous Releases, both functionalities were implemented in one AMBIENT block.

Note

The altitude calculation, now in the VEHICLE_POSITION block and based on the road slope and the vehicle speed, was incorrect in previous Releases. If you use the ambient pressure and temperature based on this calculation, the simulation behavior of your model changes after migration. The migration process automatically corrects this error.

Related topics

Basics

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

ASM Electric Components

Where to go from here

Information in this section

New Features of ASM Electric Components Blockset 3.9	42
Changes in the ASM Electric Components Demo Model	42
Migrating to ASM Electric Components Blockset 3.9	43

New Features of ASM Electric Components Blockset 3.9

TRQ_REQUEST_ COORDINATION_BEV block

Selector lever support with reverse driving and parking pawl control has been added to the block.

ELECTRIC_MACHINE_BASIC block

A new inport has been added to switch the machine on and off.
The machine can run as motor or generator, depending on the torque and engine speed direction. Therefore, the sign handling of the machine current and speed has been improved depending on the machine state.

TRQ_REQUEST_ COORDINATION block

DC link support to set the torque request to zero if the DC link is inactive has been added to the block.

Changes in the ASM Electric Components Demo Model

Vehicle Dynamics Hybrid demo model

A DC link has been added to the demo model to connect the high voltage battery with the electric machine. The soft ECU torque manager gets the start request from the start button, switches the DC link on, and controls the pre-charge phase of the DC link capacitor. If the capacitor voltage reaches the battery voltage, the pre-charge phase is finished, the DC link is ready for operation, and the battery voltage is provided to the connected electric machines.

Engine Gasoline Hybrid demo model

A DC link has been added to the demo model to connect the high voltage battery with the electric machine. The soft ECU torque manager gets the start request from the start button, switches the DC link on, and controls the pre-

charge phase of the DC link capacitor. If the capacitor voltage reaches the battery voltage, the pre-charge phase is finished, the DC link is ready for operation, and the battery voltage is provided to the connected electric machines.

Battery Electric Vehicle (BEV) demo model

In the model, the drive mode of the BEV is controlled by a selector lever. The drive mode allows forward and reverse driving, parking with parking pawl control, and a neutral mode where all torque requests are set to zero.

Migrating to ASM Electric Components Blockset 3.9

Internal adaptations

Some blocks have had internal adaptations. These have no effect on the simulation result and its functionality.

This applies to the following blocks:

- COMPRESSOR
- THREE_PHASE_INVERTER
- THREE_LEVEL_THREE_PHASE_INVERTER
- THREE_PHASE_DCM_INVERTER
- HALF_BRIDGE_INVERTER
- THREE_LEVEL_HALF_BRIDGE_INVERTER
- PMSM_CONTROLLER
- PMSM_CONTROLLER_BASIC
- STARTER_ICE

Related topics**Basics**

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

ASM Environment

Where to go from here

Information in this section

New Features of ASM Environment Blockset 4.11	44
Migrating to ASM Environment Blockset 4.11	44

New Features of ASM Environment Blockset 4.11

Geometrically optimal trajectory

With a new Python script, you can automatically calculate an optimal geometrical trajectory of a road based on the road geometry. In a graphical user interface, you can plot the calculated trajectory and then export it as free trajectory to the corresponding road in ModelDesk.

Reference speed based on driver performance envelope

There is a new `SPEED_PROFILER` block. It can be used to include the driving style characterization (driver's performance envelope) in the reference speed calculation. The driving style depends on the driver's parameters and appears while approaching a stop as well as a turn, during cornering and after leaving a turn as well as accelerating from lower to higher vehicle speeds.

ROADSTATE_RUN_TRIGGER block

This block is used for the traffic serialization feature. It provides the option to trigger the road state transitions into the run state.


MANEUVER_SCHEDULER block

The longitudinal types *stop within distance* and *final velocity* support the new *ignore driver parameters* option. If enabled, the driver parameters in the *follow road* lateral type are overridden by the longitudinal parameters.

Use the `asm_migrate_scenario` MATLAB script to perform the manual migration of maneuver and scenario MAT files created with previous dSPACE Releases.

Migrating to ASM Environment Blockset 4.11

Maneuver models

ModelDesk provides the *compatibility mode* for working with maneuvers created with Release 2017-B and earlier. Refer to [How to Prepare the ASM Model for the ModelDesk Maneuver Compatibility Mode](#) ( ASM Environment Reference).

With Release 2019-B, the compatibility mode is provided for the last time. It will be discontinued with Release 2020-A. Therefore, it is mandatory to migrate your projects from Releases 2017-B and earlier with the current Release.

GPS_POSITION block The GPS calculation for the southern hemisphere has been improved.

LANE_NETWORK block The communication vector has been updated.

ROAD block The ROAD block has been extended by several internal calculations and data structures for use with the new traffic driver feature.

Related topics

Basics

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

ASM Gasoline Engine Basic

New Features of ASM Gasoline Engine Basic Blockset 2.2.4

Support

Starting from dSPACE Release 2020-A, the ASM Gasoline Engine Basic demo model will not be available for new projects. However, the legacy support via migration for the existing models will be provided up to and including dSPACE Release 2021-A.

Use the advanced ASM Engine Gasoline or ASM InCylinder Gasoline demo models for new gasoline engine projects.

ASM Engine Testbench

ASM Engine Testbench has new functionalities:

- A list for defining parameters and values has been added to the Online Settings page. These parameters are set to the specified values when the online test bench simulation is started on a dSPACE platform.
- The Plot Selected button has been added to the Execute page to only show the selected signals.

ASM Gasoline Engine

New Features of ASM Gasoline Engine Blockset 4.0.4

HPP_CRANKBASED block

The HPP_CRANKBASED block can now handle negative values of the phi_FMU_Energized control signal.

ASM Engine Testbench

ASM Engine Testbench has new functionalities:

- A list for defining parameters and values has been added to the Online Settings page. These parameters are set to the specified values when the online test bench simulation is started on a dSPACE platform.
- The Plot Selected button has been added to the Execute page to only show the selected signals.

ASM Gasoline InCylinder

New Features of ASM Gasoline InCylinder Blockset 2.7.1

Crank-based high-pressure pump systems

The ASM InCylinder Gasoline model now supports crank-based high-pressure pump systems. With the related blocks of the ASM InCylinder Base library, the ASM Gasoline InCylinder demo systems are prepared. You can easily insert the subsystem into an ASM InCylinder Gasoline demo model.

ASM Traffic

Where to go from here

Information in this section

New Features of ASM Traffic Blockset 4.0	49
Changes in the ASM Traffic Demo Model	50
Migrating to ASM Traffic Blockset 4.0	51

New Features of ASM Traffic Blockset 4.0

Bounding box visualization The bounding boxes for objects which are detected by the Object Sensor 3D are visualized in MotionDesk. For this purpose, the object information provided by the sensor instances are merged into unique vectors by the new DETECTED_OBJECTS_MERGE block. These merged IDs and object types are used to calculate the bounding box information for each object.

In the block, you can parameterize the number of visualized boxes.

DETECTED_OBJECTS_MERGE block This block is new. It is used for the bounding box visualization.

ObjectData for Object_Sensor_3D and Object_Sensor_2D This new set of blocks provides additional information for detected objects. They include information regarding the object position in earth coordinates and geometrical information of the objects, and they provide the custom values which can be defined for objects.

COLLISION_SENSOR block With the COLLISION_SENSOR you can now handle reference points that are defined in the Traffic Object Manager. You can also define these reference points inside the simulation model.

Micro-Doppler effect simulation The Object Sensor 3D has been extended to simulate the Micro-Doppler effect of a radar. To simulate this effect, the Object Point Editor inside the Traffic Object Manager of ModelDesk lets you define rotating radar reflection points.

TRAFFIC_SCHEDULER block The ASM Traffic Scheduler has been extended by the *traffic driver* feature:

The traffic driver feature provides the following functionalities:

- Fellow vehicles and the ASM vehicle detect other vehicles on the same lane and implement a plausible follow behavior.
- Fellow vehicles and the ASM vehicle detect traffic signs and traffic lights and comply with the corresponding traffic rules.
- Fellow vehicles feature a realistic physical behavior regarding their acceleration and their speed in curves.
- Different driving styles can be specified using the new traffic driver pool objects in ModelDesk.


Different traffic driver pool objects can be assigned to the vehicles in a ModelDesk scenario. Therefore, different driving styles can be promoted for different vehicles during the simulation.

For more information on the traffic driver, refer to [Introduction to the Traffic Driver Feature](#) ([📖 ASM Traffic Guide](#))

Use the `asm_migrate_scenario` MATLAB script to perform the manual migration of traffic and scenario MAT files created with previous dSPACE Releases.

Changes in the ASM Traffic Demo Model

Parameters	<p>The ASM Traffic demo model now provides five different parameter sets in ModelDesk:</p> <ul style="list-style-type: none"> ▪ Mid-size car ▪ Van ▪ SUV ▪ Small car ▪ Luxury car
Independent XML file creation	<p>A Python script to create independent parameter XML files is now included in the ModelDesk project.</p>
Bounding box visualization	<p>The bounding box sensor visualization has been added to the MotionDesk project.</p> <ul style="list-style-type: none"> ▪ The Object Sensor 3D subsystem has been extended by the BoundingBoxData_CoorSys_E subsystem inside the ObjectData subsystem. ▪ The Traffic Animation subsystem inside the MotionDeskInterface block has been extended by a bounding box animation block.
Custom Sensor block	<p>The Custom Sensor block has been removed from the demo model. Nevertheless, the sensor is still available in the ASM Traffic library.</p>

ObjectData block	The Object Sensor 3D and ObjectSensor2D blocks have been extended by the ObjectData subsystem which provides additional information about detected objects.
COLLISION_SENSOR block	The parameterization interface for the NCAP reference points has been moved into the CollisionSensor subsystem. Additionally, now this interface includes an option to specify whether the definition in the simulation model or the one in the Traffic Object Manager must be used.
TRAFFIC_OBJECTS block	The block has been extended to handle the NCAP reference point data, that can be defined in the Traffic Object Manager in ModelDesk.
Serialization of the traffic initialization process	The initialization process of the traffic model has been changed. There is now an order of the initialization process to improve the real-time capability. For this, the maneuver run-state transition has been changed, so that the maneuver does not start until the main initialization procedures are done. Additionally, some trigger conditions for the initialization start of traffic S-functions have been changed.
Number of traffic fellows	The ASM Traffic demo is prepared for a maximum of 30 fellow vehicles. The maximum number of fellows in the demo can easily be changed via the new <code>asm_traffic_md1_update_num_fellows</code> MATLAB script.
Reference speed calculation	A new SPEED_PROFILER block is used to include the driving style characterization (driver's performance envelope) in the reference speed calculation. For more information, refer to Speed Profiler ( ASM Environment Reference).
CUSTOM_SENSOR_SCOPEZONE_CALCULATION block	The internal data handling has been improved.

Migrating to ASM Traffic Blockset 4.0

OBJECT_SENSOR_3D_CALCULATION block	<p>The block has two new inports.</p> <ul style="list-style-type: none"> ▪ One inport to enable the initialization process of the included S-function. It is used for the new serialized initialization procedure of the traffic S-functions. ▪ One inport for the fellow velocity which is required for the calculations of the new Micro-Doppler feature.
---	---

OBJECT_SENSOR_2D_CALCULATION block

The block has a new inport. The inport enables the initialization process of the included S-function. It is used for the new serialized initialization procedure of the traffic S-functions.

COLLISION_SENSOR block

The block has a new inport. The inport enables the initialization process of the included S-function. It is used for the new serialized initialization procedure of the traffic S-functions.

TRAFFIC_SCHEDULER block

The TRAFFIC_SCHEDULER block has been extended by multiple new inports and outports required for the new traffic driver feature.

Traffic objects vector

The traffic objects vector has been modified. The modifications include new parameters and descriptions for the traffic driver feature and new point data for the Micro-Doppler feature.

Related topics


Basics

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

ASM Trailer

Changes in the ASM Trailer Demo Model


Reference speed calculation

A new SPEED_PROFILER block is introduced to include the driving style characterization (driver's performance envelope) in the reference speed calculation. For more information, refer to [Speed Profiler](#) ( [ASM Environment Reference](#)).

ASM Truck

Changes in the ASM Truck Demo Model

Reference speed calculation

A new SPEED_PROFILER block is introduced to include the driving style characterization (driver's performance envelope) in the reference speed calculation. For more information, refer to [Speed Profiler](#) ( [ASM Environment Reference](#)).

ASM Vehicle Dynamics

Where to go from here

Information in this section

New Features of ASM Vehicle Dynamics Blockset 4.1.2	55
Changes in the ASM Vehicle Dynamics Demo Model	55

New Features of ASM Vehicle Dynamics Blockset 4.1.2

Adams2ASM Converter

With the Adams2ASM Converter you can convert maps for the kinematics and compliance of the wheel suspensions from Adams Car™. The Adams2ASM Converter provides Adams command files to automate the simulation in Adams Car as much as possible. The parameterization of the maps can be done via the user interface in Adams Car. Both dependent and independent suspensions are supported.

To visualize the results generated with Adams Car and to convert into the correct model structure for the ASM model, the Adams2ASM Converter provides a user interface in MATLAB. Furthermore, the results can be written directly to ModelDesk.

Changes in the ASM Vehicle Dynamics Demo Model


Creating geometrically optimal trajectories

A Python script to generate geometrically optimal trajectories is added to the ModelDesk project.

New parameter set

The ModelDesk project now contains the *Luxury car* parameter set.

Reference speed calculation

A new SPEED_PROFILER block is introduced to include the driving style characterization (driver's performance envelope) in the reference speed calculation. For more information, refer to [Speed Profiler](#) ( [ASM Environment Reference](#)).

Bus Manager (Stand-Alone)

Where to go from here

Information in this section

New Features of the Bus Manager (Stand-Alone) 6.4	57
Migrating to Bus Manager (Stand-Alone) 6.4	59

New Features of the Bus Manager (Stand-Alone) 6.4


Support of J1939

The Bus Manager (stand-alone) now supports J1939 as defined by the Society of Automotive Engineers (SAE). You can add DBC files that specify J1939-compliant IPDUs to the ConfigurationDesk application and assign J1939-compliant IPDUs with up to 8 bytes to bus configurations to configure bus communication.

For more information, refer to [Aspects of Supported CAN Bus Features](#) ( [Bus Manager \(Stand-Alone\) Implementation Guide](#)).

Improved import behavior for LDF files

The Bus Manager (stand-alone) now has an improved import behavior for LDF files. This lets you use LDF files in ConfigurationDesk applications that specify identical names for different LIN elements, such as signals and frames.

For existing ConfigurationDesk applications, you must take note of certain points when working with CAFX files. For more information, refer to [Limitations for Bus Configuration Handling](#) ( [Bus Manager \(Stand-Alone\) Implementation Guide](#)).

New and enhanced bus configuration features

The Bus Manager (stand-alone) now provides the following new bus configuration features:

Bus Configuration Feature	Available for Bus Configuration Part	Purpose
ISignal Group End-to-End Protection Status	<ul style="list-style-type: none"> ▪ Simulated ECUs ▪ Inspection 	Lets you observe the status of a received end-to-end protected ISignal group.
PDU RX Interrupt	Simulated ECUs	Lets you use RX interrupts to trigger the execution of functions in a behavior model.
SecOC Authenticator Invalidation	Manipulation	Lets you invalidate the authentication information for transmitting secured IPDUs.

Additionally, the following bus configuration features are enhanced:

- The PDU RX Status feature is now also available for the Simulated ECUs bus configuration part.
- The SecOC feature is now also available for the Inspection bus configuration part.

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

Adding bus configuration features to multi-selected, arbitrary elements

The Bus Manager (stand-alone) now lets you add bus configuration features to multi-selected, arbitrary elements. When you select multiple elements of any type in the Bus Configurations table, the Add Feature context menu command provides all bus configuration features that are available for any selected element. When you select a feature from the context menu, it is added to all selected elements for which it is valid.

Editable names of bus configuration elements

The Bus Manager (stand-alone) now lets you edit the names of the following bus configuration elements.

Communication cluster nodes that are available for elements of DBC and LDF files

When you add a DBC or LDF file to the ConfigurationDesk application, the communication cluster name is derived from the file name. When you assign elements of a DBC or LDF file to the Inspection or Manipulation part of a bus configuration, you can now specify a user-defined name for the communication cluster node in the bus configuration. This name is used in the variable description (TRC) file when you generate bus simulation containers. It lets you unambiguously access bus configuration elements via automation scripts, for example.

For more information, refer to [Basics on Bus Configurations \(Bus Manager \(Stand-Alone\) Implementation Guide\)](#).

Bus access requests By default, the names of bus access requests are derived from the elements that generate the bus access requests. You can adapt the default names according to your requirements. For more information, refer to

Basics on Bus Access Requests ([📖 Bus Manager \(Stand-Alone\) Implementation Guide](#)).

Migrating to Bus Manager (Stand-Alone) 6.4

Changed naming scheme for bus access requests The naming scheme for bus access requests has changed from Bus Manager (stand-alone) 6.3 to Bus Manager (stand-alone) 6.4. The change is the following:

Old Naming Scheme	New Naming Scheme
<Bus configuration name>: <communication cluster name> (<bus configuration part name>)	Bus access requests available for: <ul style="list-style-type: none"> ▪ Simulated ECUs, Inspection, or Manipulation bus configuration part: Bus Access Request [<bus configuration name>\<bus configuration part name>\<communication cluster name>\name of communication matrix in bus configuration] ▪ Gateways bus configuration part: Bus Access Request [<bus configuration name>\<frame gateway name>\<communication cluster name>]

If you work with automation scripts that access bus access requests via their names, you have to adapt the automation scripts if you use them with new projects and the Bus Manager (stand-alone) 6.4.

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.

ConfigurationDesk - Implementation Version

Where to go from here

Information in this section

New Features of ConfigurationDesk 6.4 (Implementation Version)	62
Migrating to ConfigurationDesk 6.4	74

New Features of ConfigurationDesk 6.4 (Implementation Version)

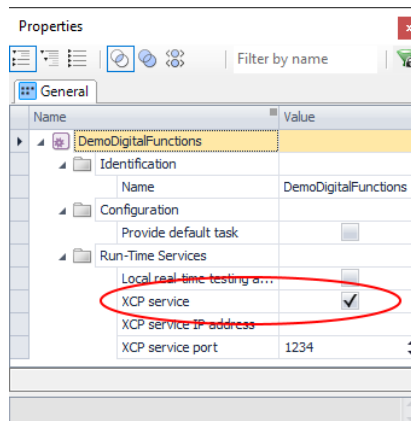
New command for replacing model implementations

ConfigurationDesk now provides the Replace Model context menu command that lets you easily replace a resolved model implementation with another model implementation. If the old model was assigned to an application process, it is removed from the application process and the new model is added to the application process. If the model interface is the same in both models, model port blocks remain resolved, and mapping lines to I/O functions and other model ports are preserved.

For more information, refer to [Replace Model](#) ( [ConfigurationDesk User Interface Reference](#)).

Configurable XCP service for Simulink models and SIC files with A2L file fragments

A Simulink model or SIC file can contain an A2L file fragment. You can assign these Simulink models/SIC files to separate application processes and configure an XCP service for them. ConfigurationDesk provides the XCP service checkbox for this purpose:



The XCP service checkbox is selected by default so that you can access the variables of the Simulink model or SIC file with the experiment software. If you clear the checkbox, you can use the relevant Simulink model or SIC file in

ConfigurationDesk without the CFD_Implementation_XCP (CFD_I_XCP) license. In this case, however, ConfigurationDesk does not create an A2L file during the build process, and you cannot access the variables of the Simulink model or SIC file with the experiment software. For more information, refer to [Specifying Options for the Build Process](#) ([📖 ConfigurationDesk Real-Time Implementation Guide](#)).

TargetLink-generated SIC files in multimodel application processes

ConfigurationDesk lets you assign Simulink implementation container files that were generated with TargetLink 5.0 to multimodel application processes. Refer to [Using Multiple Model Implementations in the Same Application Process](#) ([📖 ConfigurationDesk Real-Time Implementation Guide](#)).

Supported SIC file versions

ConfigurationDesk 6.4 supports SIC file versions as listed below:

SIC Files Created With ...	SIC Version
dSPACE Release 2019-B: <ul style="list-style-type: none"> ▪ Model Interface Package for Simulink 4.2 ▪ TargetLink 5.0 	1.7
dSPACE Release 2019-A: <ul style="list-style-type: none"> ▪ Model Interface Package for Simulink 4.1 	1.6
dSPACE Release 2018-B: <ul style="list-style-type: none"> ▪ Model Interface Package for Simulink 4.0 ▪ TargetLink 4.4 	1.5
dSPACE Release 2018-A: <ul style="list-style-type: none"> ▪ Model Interface Package for Simulink 3.6 	1.4

New BSC file features

Supported BSC file versions ConfigurationDesk 6.4 supports BSC files of version 1.7.

Support of BSC files with the PDU RX Interrupt feature of the Bus Manager You can now add BSC files with the PDU RX Interrupt feature to a ConfigurationDesk application. Refer to [Adding Bus Simulation Containers to a ConfigurationDesk Application](#) ([📖 ConfigurationDesk Real-Time Implementation Guide](#)).

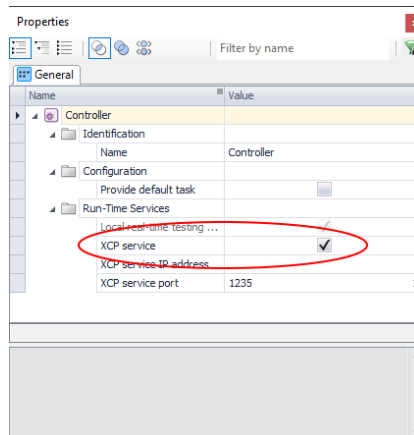
Support of BSC files in multimodel application processes ConfigurationDesk now lets you use BSC files in multimodel application processes. This is valid for BSC files with CAN/LIN communication generated by the Bus Manager. For more information, refer to [Using Multiple Model Implementations in the Same Application Process](#) ([📖 ConfigurationDesk Real-Time Implementation Guide](#)).

New V-ECU implementation features

New V-ECU implementation license required If you want to work with V-ECU implementations in the ConfigurationDesk application, you require the new CFD_I_VECU (CFD_Implementation_VECU) license. Refer to [Overview of Licenses](#) ([📖 ConfigurationDesk Real-Time Implementation Guide](#)).

Configurable XCP service for V-ECU implementations with A2L file fragments

A V-ECU implementation can contain an A2L file fragment. You can assign these V-ECU implementations to separate application processes and configure an XCP service for them. ConfigurationDesk provides the XCP service checkbox for this purpose:



The XCP service checkbox is selected by default so that you can access the variables of the V-ECU implementation with the experiment software. If you clear the checkbox, you can use the V-ECU implementation in ConfigurationDesk without the CFD_Implementation_XCP (CFD_I_XCP) license. In this case, however, ConfigurationDesk does not create an A2L file during the build process, and you cannot access the variables of the V-ECU implementation with the experiment software. For more information, refer to [Configuring the Build Process for ConfigurationDesk Applications Containing V-ECU Implementations](#) ([ConfigurationDesk Real-Time Implementation Guide](#)).

Support of V-ECU implementations that provide memory segments For application processes to which a V-ECU implementation is assigned, ConfigurationDesk creates a memory segment entry in the A2L file. This memory segment entry contains all variables supplied by the V-ECU implementation that are prepared for write access, provided that the V-ECU implementation is set to supply these variables. Refer to [Working with V-ECU Implementations That Provide Memory Segments](#) ([ConfigurationDesk Real-Time Implementation Guide](#)).

Supported V-ECU implementation container versions ConfigurationDesk 6.4 supports V-ECU implementation container versions as listed below:

V-ECU Implementations Created With...	V-ECU Implementation Version
dSPACE Release 2019-B: <ul style="list-style-type: none"> ▪ SystemDesk 5.4 ▪ TargetLink 5.0 	2.10
dSPACE Release 2019-A: <ul style="list-style-type: none"> ▪ SystemDesk 5.3 	2.9
dSPACE Release 2018-B: <ul style="list-style-type: none"> ▪ SystemDesk 5.2 ▪ TargetLink 4.4 	2.8

V-ECU Implementations Created With...	V-ECU Implementation Version
dSPACE Release 2018-A: ▪ SystemDesk 5.1	2.7

Compatible EIC files



ConfigurationDesk 6.4 supports EIC file versions as listed below:







EIC Files Created with ECU Interface Manager of ...	EIC Version
dSPACE Release 2019-B (ECU Interface Manager 2.6)	4.0.0
dSPACE Release 2019-A (ECU Interface Manager 2.5)	3.0.0
dSPACE Release 2018-B (ECU Interface Manager 2.4)	3.0.0
dSPACE Release 2018-A (ECU Interface Manager 2.3)	2.0.0
dSPACE Release 2017-B (ECU Interface Manager 2.2)	1.0.0
dSPACE Release 2017-A (ECU Interface Manager 2.1)	1.0.0
dSPACE Release 2016-B (ECU Interface Manager 2.0p1)	1.0.0

However, MicroAutoBox III systems only support EIC file version 4.0.0.

New function block types

The table below shows the new function block types:

Function Block	Description	Supported Hardware	Channel Types	Further Information
Block-Commutated PWM Out	The Block-Commutated PWM Out function block generates block-commutated PWM signals to control three-phase brushless DC (BLDC) motors.	DS6121 (SCALEXIO)	Digital Out 8	Block-Commutated PWM Out ( ConfigurationDesk I/O Function Implementation Guide)
Sine Encoder In	The Sine Encoder In function block provides access to rotary or linear analog incremental encoders that provide sinusoidal output signals. The function block can be used, for example, to measure the angular position and the speed of an electric motor.	DS6121 (SCALEXIO)	▪ Flexible In/Out 1	Sine Encoder In ( ConfigurationDesk I/O Function Implementation Guide)

Function Block	Description	Supported Hardware	Channel Types	Further Information
Hall Encoder In	The Hall Encoder In function block provides access to Hall encoders with differential and single-ended signals. The function block can be used to determine the angular position and the speed of an electric motor.	DS6121 (SCALEXIO)	<ul style="list-style-type: none"> ▪ Digital In/Out 9 ▪ Flexible In/Out 1 	Hall Encoder In ( ConfigurationDesk I/O Function Implementation Guide)
Resolver In	The Resolver In function block type provides access to resolvers to determine the angular position and the speed, for example, of an electric motor.	DS6121 (SCALEXIO)	Resolver In 2	Resolver In ( ConfigurationDesk I/O Function Implementation Guide)
EnDat Master	The EnDat Master function block provides access to absolute encoders (single-turn and multi-turn encoders) via the EnDat 2.1 or EnDat 2.2 protocol. The function block can be used to determine the angular position, the revolution count, and the speed, for example, of an electric motor.	DS6121 (SCALEXIO)	<ul style="list-style-type: none"> ▪ Flexible In/Out 1 	EnDat Master ( ConfigurationDesk I/O Function Implementation Guide)
SSI Master	The SSI Master function block provides access to absolute encoders (single-turn and multiturn encoders) that support the SSI or the BiSS-C interface. The function block can be used to determine the angular position, the revolution count, and the speed of an electric motor, for example.	DS6121 (SCALEXIO)	<ul style="list-style-type: none"> ▪ Digital In/Out 9 ▪ Flexible In/Out 1 	SSI Master ( ConfigurationDesk I/O Function Implementation Guide)
Ethernet Switch	The Ethernet Switch function block lets you configure the switching of Ethernet traffic and the characteristics of the physical layer transceivers (PHYs).	DS6333 and DS6335 (SCALEXIO)	<ul style="list-style-type: none"> ▪ Ethernet Switch 1 	Ethernet Switch ( ConfigurationDesk I/O Function Implementation Guide)
		DS1403 (MicroAutoBox III)	<ul style="list-style-type: none"> ▪ Ethernet Switch 2 	
Digital Pulse In	The Digital Pulse In function block type lets you measure the pulse duration of digital voltage signals coming from an external device.	DS1511, DS1511B1 and DS1513 (MicroAutoBox III)	Digital In 4	Digital Pulse In ( ConfigurationDesk I/O Function Implementation Guide)

Function Block	Description	Supported Hardware	Channel Types	Further Information
Voltage Signal Capture (ADC Type 4)	With the Voltage Signal Capture (ADC Type 4) function block, you can measure analog voltage signals (coming from an external device) by capturing signal sequences, for example, at configurable sample rates. The function block type is exclusively designed to be used for the ADC Type 4 module of the DS1511, DS1511B1, and DS1513 Multi-I/O Boards.	DS1511 (MicroAutoBox III) DS1511B1, DS1513 (MicroAutoBox III)	Analog In 7 Analog In 8	Voltage Signal Capture (ADC Type 4) (📖 ConfigurationDesk I/O Function Implementation Guide)
SPI Master	The SPI Master function block controls and performs a short-distance communication via the serial peripheral interface (SPI). SPI communication is a master-slave architecture with a single master.	DS1511, DS1511B1 and DS1513 (MicroAutoBox III)	Digital In 4, Digital Out 4	SPI Master (📖 ConfigurationDesk I/O Function Implementation Guide)
Non-Volatile Memory Access	The Non-Volatile Memory Access function block provides access to the non-volatile memory of the processing hardware. The function block creates a data set in the non-volatile memory and handles the data transfer between the behavior model and the non-volatile memory for this data set.	<ul style="list-style-type: none"> ▪ SCALEXIO LabBox with DS6001 ▪ SCALEXIO AutoBox with DS6001 ▪ MicroAutoBox III 	-	Non-Volatile Memory Access (📖 ConfigurationDesk I/O Function Implementation Guide)

New channel types for existing function block types

The table below shows the function blocks that support new channel types:

Function Block	New Channel Types	Supported Hardware	Further Information
Voltage In	Analog In 2	DS2680 (SCALEXIO)	Voltage In ( ConfigurationDesk I/O Function Implementation Guide)
	Analog In 5	DS6101 (SCALEXIO)	
	Analog In 16	DS6121 (SCALEXIO)	
	Analog In 7	DS1511 (MicroAutoBox III)	
	Analog In 8	DS1511B1, DS1513 (MicroAutoBox III)	
	Analog In 9	DS1513 (MicroAutoBox III)	
	Analog In 10	DS1552 (MicroAutoBox III)	
	Analog In 11	DS1552B1 (MicroAutoBox III)	
	Analog In 12	DS1552, DS1552B1 (MicroAutoBox III)	
	Analog In 14	DS1554 (MicroAutoBox III)	
Voltage Out	Analog Out 11	DS1511, DS1511B1 (MicroAutoBox III)	Voltage Out ( ConfigurationDesk I/O Function Implementation Guide)
	Analog Out 12	DS1513 (MicroAutoBox III)	
	Analog Out 13	DS1552, DS1552B1 (MicroAutoBox III)	
Multi Bit In	Digital In/Out 9	DS6121 (SCALEXIO)	Multi Bit In ( ConfigurationDesk I/O Function Implementation Guide)
	Digital In 4	DS1511, DS1511B1, DS1513 (MicroAutoBox III)	
	Digital In 5	DS1552, DS1552B1 (MicroAutoBox III)	
	Digital In/Out 6		
	Digital In/Out 8	DS1554 (MicroAutoBox III)	
Trigger In	Digital In/Out 9	DS6121 (SCALEXIO)	Trigger In ( ConfigurationDesk I/O Function Implementation Guide)
	Digital In 5	DS1552, DS1552B1 (MicroAutoBox III)	
	Trigger In 3	DS1511, DS1511B1, DS1513 (MicroAutoBox III)	

Function Block	New Channel Types	Supported Hardware	Further Information
Multi Bit Out	Digital Out 8	DS6121 (SCALEXIO)	Multi Bit Out (📖 ConfigurationDesk I/O Function Implementation Guide)
	Digital In/Out 9		
	Digital Out 4	DS1511, DS1511B1, DS1513 (MicroAutoBox III)	
	Digital Out 5	DS1552, DS1552B1 (MicroAutoBox III)	
	Digital Out 7	DS1554 (MicroAutoBox III)	
	Digital In/Out 6	DS1552, DS1552B1 (MicroAutoBox III)	
	Digital In/Out 8	DS1554 (MicroAutoBox III)	
Digital Pulse Out	Digital Out 8	DS6121 (SCALEXIO)	Digital Pulse Out (📖 ConfigurationDesk I/O Function Implementation Guide)
	Digital In/Out 9		
	Digital Out 4	DS1511, DS1511B1, DS1513 (MicroAutoBox III)	
PWM/PFM Out	Digital Out 8	DS6121 (SCALEXIO)	PWM/PFM Out (📖 ConfigurationDesk I/O Function Implementation Guide)
	Digital In/Out 9		
	Digital Out 4	DS1511, DS1511B1, DS1513 (MicroAutoBox III)	
	Digital Out 5	DS1552, DS1552B1 (MicroAutoBox III)	
	Digital Out 7	DS1554 (MicroAutoBox III)	
	Digital In/Out 6	DS1552, DS1552B1 (MicroAutoBox III)	
	Digital In/Out 8	DS1554 (MicroAutoBox III)	
Multi Channel PWM Out	Digital Out 8	DS6121 (SCALEXIO)	Multi-Channel PWM Out (📖 ConfigurationDesk I/O Function Implementation Guide)
	Digital In/Out 9		
	Digital Out 4	DS1511, DS1511B1, DS1513 (MicroAutoBox III)	

Function Block	New Channel Types	Supported Hardware	Further Information
PWM/PFM In	Digital In/Out 9	DS6121 (SCALEXIO)	PWM/PFM In (📖 ConfigurationDesk I/O Function Implementation Guide)
	Digital In 4	DS1511, DS1511B1, DS1513 (MicroAutoBox III)	
	Digital In 5	DS1552, DS1552B1, (MicroAutoBox III)	
	Digital In/Out 6	DS1552, DS1552B1, (MicroAutoBox III)	
	Digital In/Out 8	DS1554 (MicroAutoBox III)	
Digital Incremental Encoder In	Digital In/Out 9	DS6121 (SCALEXIO)	Digital Incremental Encoder In (📖 ConfigurationDesk I/O Function Implementation Guide)
	Flexible In/Out 1		
	Digital In 4	DS1511, DS1511B1 and DS1513 (MicroAutoBox III)	
Voltage Signal Capture	Analog In 16	DS6121 (SCALEXIO)	Voltage Signal Capture (📖 ConfigurationDesk I/O Function Implementation Guide)
SENT In	Digital In 4	DS1511, DS1511B1 and DS1513 (MicroAutoBox III)	SENT In (📖 ConfigurationDesk I/O Function Implementation Guide)
CAN	CAN 3	DS1511 and DS1511B1 (MicroAutoBox III)	CAN (📖 ConfigurationDesk I/O Function Implementation Guide)
	CAN 4	DS1513 (MicroAutoBox III)	
	CAN 5	DS4342 (MicroAutoBox III)	
FlexRay	FlexRay 3	DS4340 (MicroAutoBox III)	FlexRay (📖 ConfigurationDesk I/O Function Implementation Guide)
LIN	LIN 3	DS1511, DS1511B1 and DS1513 (MicroAutoBox III)	LIN (📖 ConfigurationDesk I/O Function Implementation Guide)
System Shutdown	–	<ul style="list-style-type: none"> ▪ SCALEXIO LabBox with DS6001 ▪ SCALEXIO AutoBox with DS6001 ▪ MicroAutoBox III 	System Shutdown (📖 ConfigurationDesk I/O Function Implementation Guide)
Power On Signal In	–	<ul style="list-style-type: none"> ▪ SCALEXIO LabBox with DS6001 ▪ SCALEXIO AutoBox with DS6001 ▪ MicroAutoBox III 	Power On Signal In (📖 ConfigurationDesk I/O Function Implementation Guide)

New features of the Bus Manager

Support of J1939 The Bus Manager now supports J1939 as defined by the Society of Automotive Engineers (SAE). You can add DBC files that specify J1939-compliant IPDUs to the ConfigurationDesk application and assign J1939-compliant IPDUs with up to 8 bytes to bus configurations to configure bus communication.

For more information, refer to [Aspects of Supported CAN Bus Features](#) ([📖 ConfigurationDesk Bus Manager Implementation Guide](#)).

Improved import behavior for LDF files The Bus Manager now has an improved import behavior for LDF files. This lets you use LDF files in ConfigurationDesk applications that specify identical names for different LIN elements, such as signals and frames.

For existing ConfigurationDesk applications, you must take note of certain points when working with CAFX files. For more information, refer to [Limitations for Bus Configuration Handling](#) ([📖 ConfigurationDesk Bus Manager Implementation Guide](#)).

New and enhanced bus configuration features The Bus Manager now provides the following new bus configuration features:

Bus Configuration Feature	Available for Bus Configuration Part	Purpose
ISignal Group End-to-End Protection Status	<ul style="list-style-type: none"> ▪ Simulated ECUs ▪ Inspection 	Lets you observe the status of a received end-to-end protected ISignal group.
PDU RX Interrupt	Simulated ECUs	Lets you use RX interrupts to trigger the execution of functions in a behavior model.
SecOC Authenticator Invalidation	Manipulation	Lets you invalidate the authentication information for transmitting secured IPDUs.

Additionally, the following bus configuration features are enhanced:

- The PDU RX Status feature is now also available for the Simulated ECUs bus configuration part.
- The SecOC feature is now also available for the Inspection bus configuration part.

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

Adding bus configuration features to multi-selected, arbitrary elements The Bus Manager now lets you add bus configuration features to multi-selected, arbitrary elements. When you select multiple elements of any type in the Bus Configurations table, the Add Feature context menu command provides all bus configuration features that are available for any selected element. When you select a feature from the context menu, it is added to all selected elements for which it is valid.

Editable names of communication cluster nodes The Bus Manager now lets you edit the names of communication cluster nodes that are available for elements of DBC and LDF files.

When you add a DBC or LDF file to the ConfigurationDesk application, the communication cluster name is derived from the file name. When you assign elements of a DBC or LDF file to the Inspection or Manipulation part of a bus configuration, you can now specify a user-defined name for the communication cluster node in the bus configuration. This name is used in the variable description (TRC) file when you build a real-time application or generate bus simulation containers. It lets you unambiguously access bus configuration elements via automation scripts, for example.

For more information, refer to [Basics on Bus Configurations](#) ([📖 ConfigurationDesk Bus Manager Implementation Guide](#)).

Editable names of bus access requests The Bus Manager now lets you edit the names of bus access requests. By default, the names of bus access requests are derived from the elements that generate the bus access requests. You can adapt the default names according to your requirements. For more information, refer to [Basics on Bus Access Requests](#) ([📖 ConfigurationDesk Bus Manager Implementation Guide](#)).

New features concerning hardware support

ConfigurationDesk supports the following new SCALEXIO hardware:

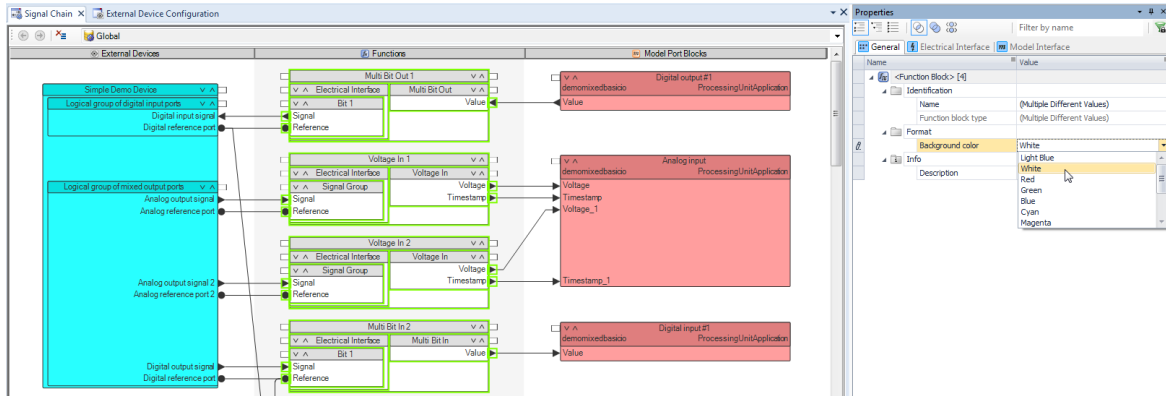
- DS6601 and DS6602 FPGA base boards
New SCALEXIO FPGA base boards with the Xilinx® Kintex® Ultrascale KU035 FPGA (DS6601) and Xilinx Kintex Ultrascale+ KU15P FPGA (DS6602). The new FPGA base boards are compatible with the DS2655 FPGA base boards.
- DS6121 Multi-I/O Board
The DS6121 Multi-I/O Board is a single-slot SCALEXIO I/O board that provides the required interfaces for electric motor control applications. It can be used for up to two electric motors.
- DS6321 UART Board
The DS6321 UART Board provides four independent UART channels, each supporting RS232, RS422, RS485, or K-Line at a time.

ConfigurationDesk supports the following hardware components of the new MicroAutoBox III:

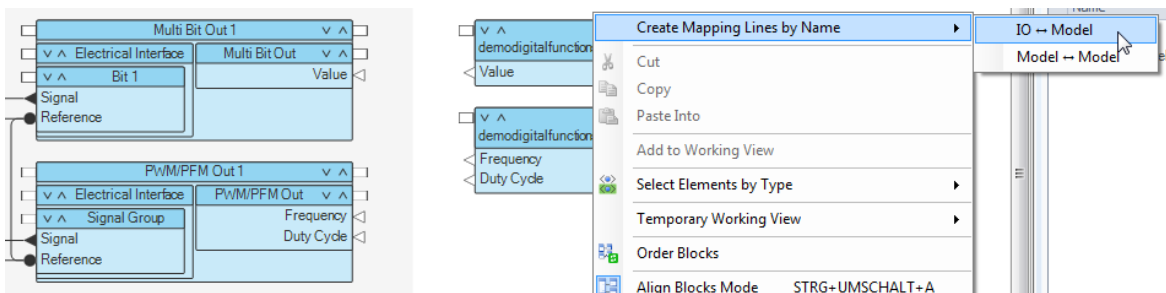
- DS1403 Processor Board
- DS1511 Multi I/O Board
- DS1511B1 Multi I/O Board
- DS1513 Multi I/O Board
- DS1514 FPGA Base Board
- DS1552 Multi I/O Board
- DS1552B1 Multi I/O Board
- DS1553 AC Motor Control I/O Module
- DS1554 Engine Control I/O Module
- DS4340 FlexRay Interface Module
- DS4342 CAN FD Interface Module

User interface improvements

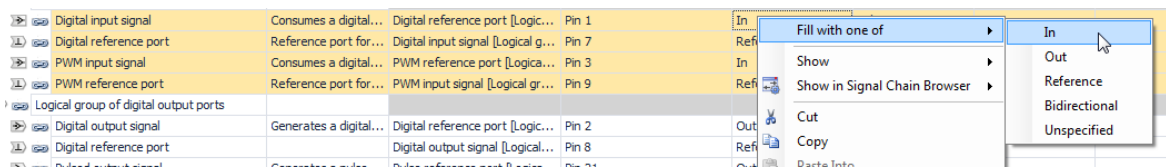
- You can now change the background color of selected blocks.



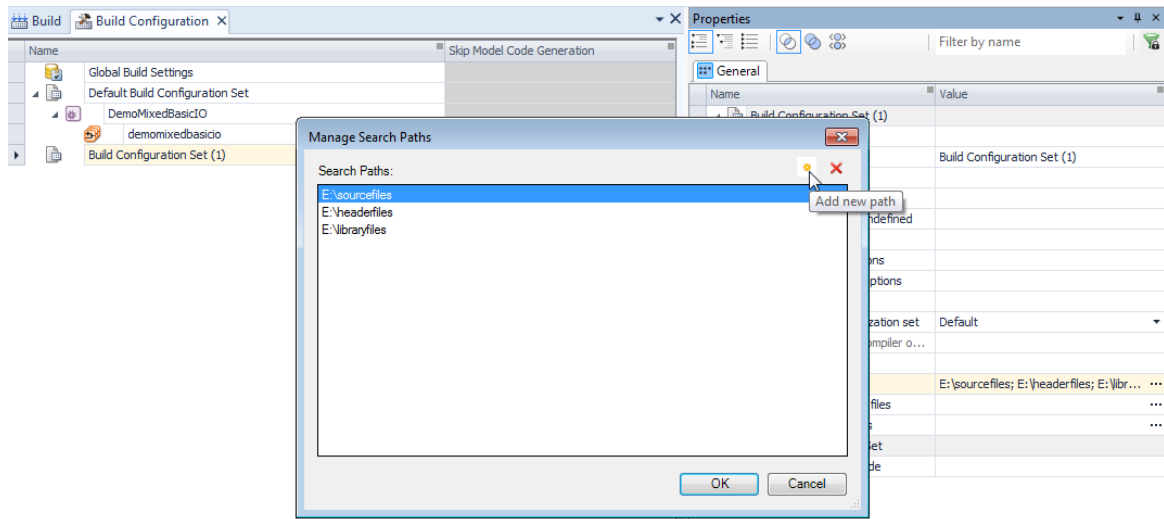
- You can now easily map function and model ports of selected blocks or in selected working views in one step if the port names match.



- You can now map ports in the signal chain by clicking one port, pressing **Ctrl+Alt**, and afterwards clicking another port.
- You can now select multiple blocks in a working view column by clicking a block, pressing **Shift**, and afterwards clicking a different block. This way, you select both blocks and all blocks between them in the column.
- The Fill With functionality was extended so that if a list of values is available for a property, you can now use the Fill with one of command to select the same property value for multiple elements from the list of available property values.



- You can now use dialogs to easily add search paths, custom source files, and custom libraries to build configuration sets.



New features of the tool automation interface

ConfigurationDesk's automation interface supports additional features of ConfigurationDesk. For more information, refer to [New Features and Changes to the Automation Interface for Release 2019-B](#) ([ConfigurationDesk Automating Tool Handling](#)).

Migrating to ConfigurationDesk 6.4

Discontinuation of Python 2.7

The support of Python 2.7 was discontinued with dSPACE Release 2018-B. Python 3.6 is now supported.

You can find information on changes and migration aspects of Python scripts in dSPACE products on the dSPACE website. Refer to <http://www.dspace.com/go/Python36Migration>.

Note

Python scripts that have been added to a ConfigurationDesk project in a previous ConfigurationDesk version via Insert Script or Import Script are automatically converted to Python 3.6 when you open the project. The script migration cannot be reverted.

Discontinuation of SCALEXIO Ethernet Solution

The SCALEXIO Ethernet Solution is discontinued as follows:

- The end-of-life date is January 31, 2021. You can still buy the product up to and including January 31, 2019.

- New Releases of the SCALEXIO Ethernet Solution will still be available for customers with a Software Maintenance Service contract until at least January 31, 2020.
- Customers with a Software Maintenance Service contract who work with dSPACE Release 2018-B will be automatically migrated to the new ConfigurationDesk UDP/TCP function blocks.

For new projects (using dSPACE Release 2018-A and later), we recommend that you use the new UDP/TCP function blocks that are natively integrated in ConfigurationDesk. They provide additional and new options such as IPv6, UDP Multicast support, and enhanced TCP status information.

Note: The dedicated license is required for using the new UDP/TCP function blocks in ConfigurationDesk.

FPGA custom function blocks with APU functionality

As of dSPACE Release 2018-B, the angle range handling of the angular processing unit (APU) was changed. FPGA custom function blocks that use the APU in the 360° angle range are incompatible if they are built with the FPGA Programming Blockset 3.5 or earlier.

To resolve the incompatibility, use the FPGA model/code of the incompatible FPGA custom function block and build a new FPGA custom function block with the RTI FPGA Programming Blockset 3.6 or later. The RTI FPGA Programming Blockset automatically migrates the framework of the FPGA model/code to the current version.

Changed naming scheme for bus access requests (available for Bus Manager elements)

The naming scheme for bus access requests has changed from ConfigurationDesk 6.3 to ConfigurationDesk 6.4. The change is the following:

Old Naming Scheme	New Naming Scheme
<Bus configuration name>: <communication cluster name> (<bus configuration part name>)	Bus access requests available for: <ul style="list-style-type: none"> ▪ Simulated ECUs, Inspection, or Manipulation bus configuration part: Bus Access Request [<bus configuration name>\<bus configuration part name>\<communication cluster name>\name of communication matrix in bus configuration] ▪ Gateways bus configuration part: Bus Access Request [<bus configuration name>\<frame gateway name>\<communication cluster name>]

If you work with automation scripts that access bus access requests via their names, you have to adapt the automation scripts if you use them with new ConfigurationDesk projects.

ControlDesk

Where to go from here

Information in this section

New Features of ControlDesk 7.1	78
Migrating to ControlDesk 7.1	84

New Features of ControlDesk 7.1

Where to go from here

Information in this section

New Project and Experiment Features (ControlDesk 7.1)	78
Gives an overview of new features for handling projects and experiments in ControlDesk 7.1.	
New Features of Platform Management and Platforms/Devices (ControlDesk 7.1)	79
Gives an overview of new features of platform management and platforms/devices in ControlDesk 7.1.	
New Instrument Features (ControlDesk 7.1)	79
Gives an overview of the new instrument features in ControlDesk 7.1.	
New Measurement and Recording Features (ControlDesk 7.1)	80
Gives an overview of the new measurement and recording features in ControlDesk 7.1.	
New Bus Navigator Features (ControlDesk 7.1)	80
Gives an overview of the new Bus Navigator features in ControlDesk 7.1.	
Additional Enhancements and Changes with ControlDesk (ControlDesk 7.1)	83
Gives an overview of additional enhancements as of ControlDesk 7.1.	

New Project and Experiment Features (ControlDesk 7.1)

Saving a project in another project root directory

ControlDesk now lets you save a project in another project root directory.


Refer to [Save As / Rename Dialog](#) ( [ControlDesk Project and Experiment Management](#)).

Related topics

References


[Save As / Rename Dialog](#) ( [ControlDesk Project and Experiment Management](#))

New Features of Platform Management and Platforms/Devices (ControlDesk 7.1)

New platform to support MicroAutoBox III ControlDesk features the MicroAutoBox III platform to support the new MicroAutoBox III.
Refer to [MicroAutoBox III Platform](#) ( [ControlDesk Platform Management](#)).

SCALEXIO: Support of DS6121 Multi-I/O Board and DS6321 UART Board ControlDesk now supports the following new SCALEXIO boards:

- DS6121 Multi-I/O Board
- DS6321 UART Board

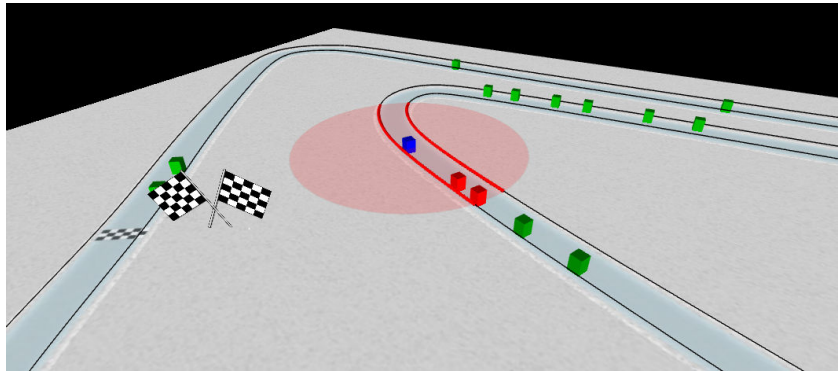
SCALEXIO: Refreshing the display of client processes ControlDesk now lets you refresh the display of the client processes that access the selected unit.
Refer to [Show Connected Clients](#) ( [ControlDesk Platform Management](#)).

Ethernet Bus Monitoring device: Support for signals of TCP Ethernet PDUs The ControlDesk Ethernet Bus Monitoring device now supports signals of TCP Ethernet PDUs.
You can do the following:

- Visualize signals of TCP Ethernet PDUs in ControlDesk instruments.
- Measure signals of TCP Ethernet PDUs, and include them in recordings.

New Instrument Features (ControlDesk 7.1)

New 3-D Viewer The new 3-D Viewer lets you display items in a 3-D environment. You can display stationary items or connect the properties of items to variables to visualize changing item properties.
The following illustration shows the 3-D Viewer. It displays a simplified race track scenario with an ego-vehicle (blue) and other vehicles inside (red) and outside (green) the fixed range of an ego-vehicle sensor.



- For a demo introducing the 3-D Viewer, refer to [Instrumentation Demo](#) ([📖 ControlDesk Introduction and Overview](#)).
- For tutorial videos that show the handling of the 3-D Viewer, refer to [Tutorial Videos for ControlDesk](#) ([📖 ControlDesk Introduction and Overview](#)).

For more information, refer to [Basics of Handling the 3-D Viewer](#) ([📖 ControlDesk Instrument Handling](#)).

New Measurement and Recording Features (ControlDesk 7.1)

Importing GPX files

You can now import positioning data, e.g., the latitude, longitude, and elevation, in the GPS Exchange (GPX) format to a ControlDesk experiment, and open the GPX file in the Measurement Data Pool. This allows you to visualize recorded positioning data in ControlDesk instruments.

Refer to [Import \(Measurement Data File\)](#) ([📖 ControlDesk Measurement and Recording](#)).

New Bus Navigator Features (ControlDesk 7.1)

Reactivating FlexRay Bus Instrument support

For real-time applications configured with the dSPACE FlexRay Configuration Package, dSPACE reactivates the following FlexRay-related features, which were discontinued as of ControlDesk 7.0:

- Display of FlexRay bus configuration structures in the Bus Navigator tree.
- Creating Bus Instruments for FlexRay.




dSPACE will provide a patch for ControlDesk 7.0 and 7.1. Later ControlDesk versions will provide these features automatically.

Support of new and enhanced bus configuration features in Bus Manager applications

The Bus Navigator supports the following new and enhanced bus configuration features in Bus Manager applications:





Secure Onboard Communication (SecOC) The Bus Navigator now supports displaying and invalidating authentication information on secured CAN PDUs.

Refer to:

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



End-to-End Protection (E2E) The Bus Navigator now supports displaying information on the status of end-to-end protected signal groups.

Refer to:

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

RX Status The Bus Navigator now supports displaying the status of an RX PDU that is assigned to the simulated ECUs part of a bus configuration.

Refer to:

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

RX Interrupt The Bus Navigator now supports enabling and disabling the CAN PDU RX Interrupt feature.

Refer to:

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

Ethernet Bus Monitoring device: Display of Ethernet PDUs and signals

The Bus Navigator tree now displays Ethernet PDUs and signals for the bus configuration of an Ethernet Bus Monitoring device to which a variable description was added.

You can also generate RX layouts for PDUs with signals. The following illustration shows an example instrument:

Bus Navigator Array: Ethernet_IPv4_IPv6_Multicast_Pdus_AUTOSAR4_3 - PDU_41_42_43			
Connection Details			
VLAN ID	100	Destination Address	239.255.1.100
Source Address	192.1.41.100	Destination Port	20000
Source Port	20000	Header ID	1234
Protocol	UDP		
Signals			
Signal Name	Signal Value (Source)	Signal Value (Converted)	
PDU_41_42_43_Signal	---	---	
Ethernet_IPv4_IPv6_Multicast_Pdus_AUTOSAR4_3 - PDU_41_42_43			

Refer to [Bus Instrument \(RX Type for Ethernet\)](#) ([ControlDesk Bus Navigator](#)).

LIN bus statistics

The ControlDesk Bus Navigator now lets you display LIN bus statistics during bus monitoring.

The following illustration shows an example:

Identifier	Monitor Time	ID	Name	Contr
0x0A [0x1, Tx]	186.4583330	0x00A	---	1
0x0B [0x1, Rx]	186.4683330	0x00B	---	1
0x0C [0x1, Rx]	186.4783330	0x00C	---	1
0x0A [0x1, Tx]	186.4783330	0x00A	---	1
0x0B [0x1, Rx]	186.4883330	0x00B	---	1
0x0C [0x1, Rx]	186.4933330	0x00C	---	1
0x0A [0x1, Tx]	186.4983330	0x00A	---	1
0x0B [0x1, Rx]	186.5083330	0x00B	---	1
0x0C [0x1, Rx]	186.5133330	0x00C	---	1
0x0A [0x1, Tx]	186.5183330	0x00A	---	1
0x0B [0x1, Rx]	186.5283330	0x00B	---	1
0x0C [0x1, Rx]	186.5333330	0x00C	---	1
0x0A [0x1, Tx]	186.5383330	0x00A	---	1
0x0B [0x1, Rx]	186.5483330	0x00B	---	1
0x0C [0x1, Rx]	186.5533330	0x00C	---	1
0x0A [0x1, Tx]	186.5583330	0x00A	---	1
0x0B [0x1, Rx]	186.5683330	0x00B	---	1
0x0C [0x1, Rx]	186.5733330	0x00C	---	1
0x0A [0x1, Tx]	186.5783330	0x00A	---	1
0x0B [0x1, Rx]	186.5883330	0x00B	---	1
0x0C [0x1, Rx]	186.5933330	0x00C	---	1

Type	Value	Min	Max
Statistic Application Time	850.7550000	---	---
Busload (Average)	23.00 %	23.00 %	23.00 %
Busload (Peak)	23.00 %	23.00 %	23.00 %
Total Frames	27858	---	---
Frames per Second	150	138	150
Bus Errors	0	---	---
Cycle Time [ms]	20	20	20
Detected Response Errors	0	---	---
Diag No Response	0	---	---
ETF Collisions	0	---	---
ETF RX Response Errors	0	---	---
ETF RX Responses	0	---	---
ETF TX Response Errors	0	---	---
ETF TX Responses	0	---	---
Header Duration [ms]	---	---	---
Header Tolerance	---	---	---
Master Init Duration [ms]	---	---	---
Measured Baud Rate	19200	---	---
Resolved ETF Collisions	0	---	---
Response Duration [ms]	---	---	---

Displaying LIN bus statistics is supported for LIN controller nodes of a SCALEXIO or VEOS LIN channel.

Refer to [How to Display Bus Statistics](#) ([ControlDesk Bus Navigator](#)).

Additional Enhancements and Changes with ControlDesk (ControlDesk 7.1)

New tutorial videos for the Internal Interpreter

The dSPACE website provides new tutorial videos that show you how to work with the ControlDesk Internal Interpreter. Refer to:

https://www.dspace.com/go/tutorial_cd_internal_interpreter

Migrating to ControlDesk 7.1

Where to go from here

Information in this section

Discontinuations in ControlDesk	84
Gives you an overview of the discontinuations in ControlDesk.	
Migrating to ControlDesk 7.1	86
To migrate from ControlDesk 7.0 to ControlDesk 7.1 and reuse existing experiments, you might have to carry out the following migration steps.	

Discontinuations in ControlDesk

Discontinuations as of ControlDesk 7.1

FXF file support As of ControlDesk 7.1, FXF files for exchanging formulas of calculated variables are no longer supported.

However, as of ControlDesk 7.1, you can exchange formulas by exporting/importing them to/from the VXF file format.

For information on exchanging formulas, refer to [How to Export and Import Template Formulas for Calculated Variables](#) ([📖 ControlDesk Variable Management](#)).

For migration information, refer to [Migrating from ControlDesk 7.0 to 7.1](#) ([📖 ControlDesk Introduction and Overview](#)).

DCM and DSV file support As of ControlDesk 7.1, the import and export of the following file formats for data sets is no longer supported:

- BOSCH DCM file format
- dSPACE Cal Data (DSV) file format

Use the CDFX file format, which is the standard ControlDesk file format for data set files. Refer to [Import Data Set\(s\)](#) ([📖 ControlDesk Calibration and Data Set Management](#)) and [Export Data Sets / Export](#) ([📖 ControlDesk Calibration and Data Set Management](#)).

During experiment migration, DSV and DCM files are deleted.

Taking snapshots As of ControlDesk 7.1, you can no longer take snapshots, i.e., read the current values of variable values defined in a label list, and save them to a CSV file.

Defining calculated variables based on recorded signals As of ControlDesk 7.1, it is no longer possible to define calculated variables using recorded signals as input signals in the **Measurement Data Pool** controlbar.

As an alternative, you can do the following:

1. Define calculated variables *before* a recording in the **Variables** controlbar.
2. Include the variables in the recording.

During experiment migration, calculated variables using recorded signals as input signals in the Measurement Data Pool controlbar are deleted.

Discontinued bookmark types As of ControlDesk 7.1, the following bookmark types are no longer supported:

Bookmark Type in the User Interface	Discontinued Constant/Enumeration	
	Measurement Data API	Tool Automation
Failure pattern/pin activated ¹⁾	btFailureSimulationActivate	FailureSimulationActivate
Failure pattern/pin deactivated ²⁾	btFailureSimulationDeactivate	FailureSimulationDeactivate
Warning of the DAQ processor ³⁾	btDAQProcessorWarning	_3)

- ¹⁾ The related On failure simulation activate condition for triggered recording is no longer supported either.
- ²⁾ The related On failure simulation deactivate condition for triggered recording is no longer supported either.
- ³⁾ This bookmark type was unavailable via the user interface and via tool automation.

For migration aspects, refer to [Experiments containing 'Failure pattern/pin \(de\)activated' bookmarks and recording trigger conditions](#) on page 87.

Discontinuations in ControlDesk (dSPACE Release 2020-A and later)

IDF file import As of dSPACE Release 2020-A, importing files in the IDF format will no longer be supported.

Use the ASAM MDF 4.1 file format (file name extension: MF4), which is the standard ControlDesk file format for measurement data files.

DS1005 PPC Board platform As of dSPACE Release 2020-A, the DS1005 PPC Board platform is no longer available, and the DS1005 is no longer supported.

For new projects, we recommend that you use the DS1007 PPC Processor Board.

DCI-GSI1 device As of dSPACE Release 2020-A, the DCI-GSI1 device is no longer available, and the DCI-GSI1 is no longer supported.

For new projects, we recommend that you use the DCI-GSI2.

DCI-CAN1 device As of dSPACE Release 2020-A, the DCI-CAN1 device is no longer available, and the DCI-CAN1 is no longer supported.

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

Support for MicroAutoBox II with DS1512 As of dSPACE Release 2020-A, ControlDesk no longer supports MicroAutoBox II with its variants 1401/1511/1512 and DS1401/1512/1513.

For new projects, we recommend that you use the MicroAutoBox II or MicroAutoBox III variants with DS1514 I/O Board.

Global data sets As of dSPACE Release 2020-A, ControlDesk no longer supports global data sets, i.e., you can no longer make a data set available to all the experiments of the current ControlDesk project.

Putting projects under version control from within ControlDesk As of dSPACE Release 2020-A, you can no longer put projects under version control from within ControlDesk, i.e., ControlDesk will no longer provide commands to put a project under version control.

End of software support for discontinued dSPACE hardware

For information on the end of software support for discontinued dSPACE hardware, refer to [Discontinuations](#) on page 17.

Related topics

Basics

[Migrating from ControlDesk 7.0 to 7.1](#) ([📖 ControlDesk Introduction and Overview](#))

Migrating to ControlDesk 7.1

Introduction

To migrate from ControlDesk 7.0 to ControlDesk 7.1 and reuse existing experiments, you might have to carry out the following migration steps.

Note

To migrate to ControlDesk 7.1 from versions earlier than 7.0, you might also have to perform the migration steps of the intervening ControlDesk versions.

For more information, refer to [Migrating from Prior Versions of ControlDesk](#) ([📖 ControlDesk Introduction and Overview](#)).

Migrating experiments named "BusNavigator"

When you open a project last saved with ControlDesk 7.0 or earlier in ControlDesk 7.1, automatic project migration fails if the project contains an experiment named "BusNavigator".

To solve the problem, rename the experiment in an earlier version of ControlDesk.

For more information, refer to [Problem when Migrating Projects](#) ([📖 ControlDesk Project and Experiment Management](#)).

GNSS device: Modified variable name

In ControlDesk 7.1, the name of the `Speed` variable of the GNSS device has been modified:


ControlDesk Version	Variable Name
Up to and including ControlDesk7.0	'GNSS()://Speed/Speed [km/h]'
As of ControlDesk 7.1	'GNSS()://Speed/Speed [km h]'

As a consequence, the following migration aspects apply:

- You have to adapt calculated variables that use the **Speed** variable as an input signal.
- You have to adapt automation scripts that use the **Speed** variable.

Ethernet Bus Monitoring device: Modified raster name

In ControlDesk 7.1, the measurement raster name of the variables of the Ethernet Bus Monitoring device has been modified.

As a result, variables of the Ethernet Bus Monitoring device that are on the measurement signal list but not visualized in an instrument, i.e., that are indicated with the  symbol in the measurement signal list, are removed from the list as soon as you reload (refer to Reload (Variable Description) ([ControlDesk Variable Management](#))) the variable description.

You can do the following:

- Place the variables on an instrument *before* you reload the variable description, or
- Add the variables to the measurement signal list (refer to Add to Measurement Signal List ([ControlDesk Variable Management](#))) again *after* you reload the variable description.

Experiments containing 'Failure pattern/pin (de)activated' bookmarks and recording trigger conditions

As of ControlDesk 7.1, the following bookmark types are no longer supported:

Bookmark Type in the User Interface	Discontinued Constant/Enumeration	
	Measurement Data API	Tool Automation
Failure pattern/pin activated ¹⁾	btFailureSimulationActivate	FailureSimulationActivate
Failure pattern/pin deactivated ²⁾	btFailureSimulationDeactivate	FailureSimulationDeactivate
Warning of the DAQ processor ³⁾	btDAQProcessorWarning	_3)

¹⁾ The related On failure simulation activate condition for triggered recording is no longer supported either.

²⁾ The related On failure simulation deactivate condition for triggered recording is no longer supported either.

³⁾ This bookmark type was unavailable via the user interface and via tool automation.

Automatic bookmark type migration when opening measurement data files When you open a measurement data file that contains bookmarks of the discontinued types in ControlDesk 7.1 or later, the types are changed to *Manual* automatically. All the other bookmark properties remain the same.

Removal of predefined conditions from recording trigger rules When you open an experiment that contains a trigger rule for triggered recording, and if the trigger rule contains an On failure simulation activate or On failure simulation deactivate predefined condition, the condition is removed from the trigger rule definition. This applies also when you import a trigger rule file.

Measurement Data API changes

As of ControlDesk 7.1, the following BookmarkTypeConstants are no longer available:

- btFailureSimulationActivate
- btFailureSimulationDeactivate

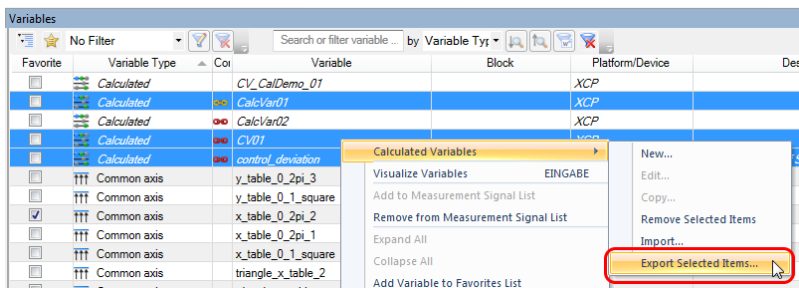
This is due to the discontinuation of the 'Failure pattern/pin (de)activated' bookmark types (refer to Discontinuations as of ControlDesk 7.1 ([ControlDesk Introduction and Overview](#))).

Migrating formulas for calculated variables

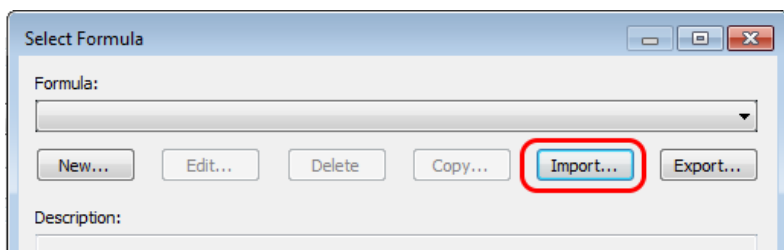
As of ControlDesk 7.1, FXF files for exchanging formulas of calculated variables are no longer supported.

If you want to reuse (or exchange) formulas from ControlDesk 7.0 or earlier you must use the VXF file format. Perform the following steps:

1. In ControlDesk 7.0 or earlier, make sure that each formula that you want to reuse (or exchange) is assigned to a calculated variable.
2. In the Variables controlbar, select the calculated variables and export them to a VXF file.



3. In ControlDesk 7.1 or later, import the resulting VXF file via the Import button of the Select Formula dialog.




You can open the Select Formula dialog via the Properties for Calculated Variables dialog. Refer to [How to Export and Import Template Formulas for Calculated Variables](#) ([ControlDesk Variable Management](#)).

Migrating from earlier ControlDesk versions

To migrate from earlier 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)

dSPACE FlexRay Configuration Package

New Features of dSPACE FlexRay Configuration Package 4.4

FlexRay Configuration Package

Support of MicroAutoBox III The FlexRay Configuration Package supports the MicroAutoBox III as the new dSPACE platform.

To set up a FlexRay network for real-time simulation with the MicroAutoBox III, you can use the dSPACE FlexRay Configuration Package consisting of the following components:

- FlexRay Configuration Tool
- FlexRay Configuration Blockset
- FlexRay function block type in ConfigurationDesk

Refer to [FlexRay Configuration with the FlexRay Configuration Tool](#) ( FlexRay Configuration Tool Guide).

dSPACE Python Extensions

New Features of dSPACE Python Extensions 3.2

New features

Python Extensions 3.2 does not have new features.

Support of Python 2.7 was discontinued as of dSPACE Release 2018-B. Python 3.6 is now supported.

You find information on changes and migration aspects of Python scripts in dSPACE products on the dSPACE website. Refer to <http://www.dspace.com/go/Python36Migration>.

Note

If you want to use the `matlablib2` Python module with MATLAB R2019a, you need MATLAB R2019a Update 5 or later. For more information, refer to [Supported MATLAB Releases](#) on page 219.

dSPACE Installation Manager

Where to go from here

Information in this section

New Features of dSPACE Installation Manager 5.4	95
Migrating to dSPACE Installation Manager 5.4	96

New Features of dSPACE Installation Manager 5.4

New features for managing licenses

Downloading license files for dSPACE Releases earlier than dSPACE Release 2017-B

If you own a software product from dSPACE Release 2017-B or later and you want to install and use a version of this product from a dSPACE Release earlier than 2017-B, you must have legacy license files and a CodeMeter CmContainer with activated licenses. Until now, you had to contact dSPACE and provide specific information before dSPACE sent you the license files.

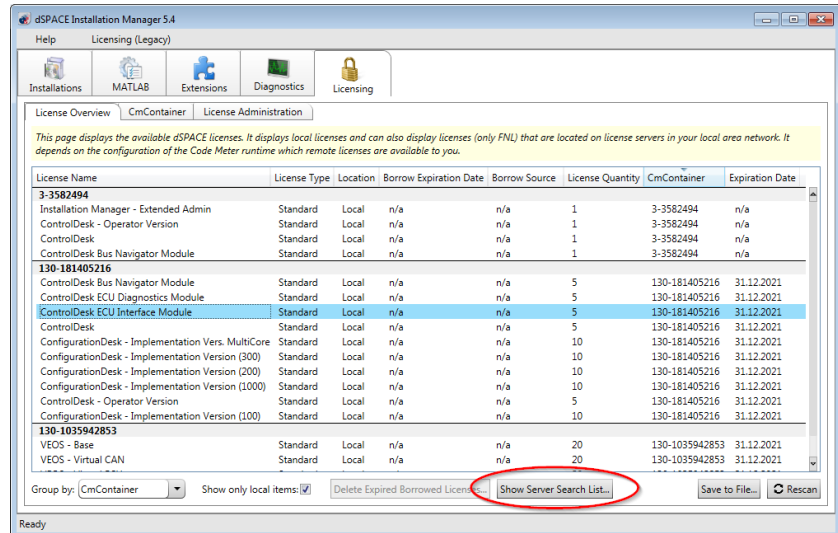
As of October 2019, you can use dSPACE Installation Manager 5.4 to download license files specifically prepared on the basis of the licenses you purchased.

Legacy licensing using CodeMeter licenses maps former product versions to an available license so that you can install and use products from dSPACE Release 7.4 (2012-B) up to and including dSPACE Release 2017-A.

Refer to [Legacy Licensing Using CodeMeter Licenses](#) ([Working with CodeMeter Licensing Technology](#)).

Request of ticket information from local CmContainer You can now request ticket information from a local CmContainer, for example, a connected CmDongle. In this case, you can have dSPACE Installation Manager read and collect all ticket information related to the licenses that are available on the CmContainer. Refer to [How to Request and Show Ticket Information](#) ([Working with CodeMeter Licensing Technology](#)).

Accessing the CodeMeter WebAdmin server search list You can now access the server search list in CodeMeter WebAdmin directly via the dSPACE Installation Manager. Click the Show Server Search List button on the License Overview page. Refer to the following illustration:



Support of CSV files You can now save displayed ticket information or displayed license information to a CSV file. This also applies for the file import.

Changes to license borrowing

- You can now borrow floating network licenses from a CmContainer on a dSPACE License Server to a CmDongle. The limitation of dSPACE Release 2019-A no longer applies.
- You can now borrow licenses for a period shorter than the maximum borrow period so that the licenses will automatically expire in the borrow container and become available on the server again after the period you specify.

Refer to [License Borrowing \(Working with CodeMeter Licensing Technology\)](#).


Migrating to dSPACE Installation Manager 5.4

Using CmDongles

If you want to work with licenses on CmDongles in combination with dSPACE Installation Manager 5.4, for example, to activate, deactivate, or update licenses, the dongles must have firmware version 4.10.

To use CmDongles shipped for Releases earlier than dSPACE Release 2019-A, a firmware update is required. CmDongles shipped for dSPACE Release 2019-A and later contain the required firmware version.

dSPACE Installation Manager checks if the firmware of a connected dongle matches the required firmware version and displays if an update is necessary.

For instructions on updating the firmware, refer to [How to Update the Firmware of a CmDongle](#) ( [Working with CodeMeter Licensing Technology](#)).

dSPACE XIL API .NET

New Features of dSPACE XIL API .NET 2019-B

New features

The dSPACE XIL API.NET 2019-B has the following new feature:

- dSPACE XIL API MAPIPort supports MicroAutoBox III as the new dSPACE platform.

For information on the Python and XIL API support changed with dSPACE Release 2018-B, refer to <http://www.dspace.com/go/Python36Migration> and the *New Features and Migration* document from dSPACE Release 2018-B.

ECU Interface Manager

Where to go from here

Information in this section

New Features of ECU Interface Manager 2.6	101
An overview of the new features of ECU Interface Manager 2.6.	
Compatibility of ECU Interface Manager 2.6	102
Provides information on the compatibility of ECU Interface Manager 2.6.	
Migrating to ECU Interface Manager 2.6	103
Information on how to migrate to ECU Interface Manager 2.6.	

New Features of ECU Interface Manager 2.6

Support of MicroAutoBox III

The ECU Interface Manager lets you prepare external ECU interfacing using ConfigurationDesk and the new MicroAutoBox III:

- You can configure the access of the MicroAutoBox III to ECU functions.
- You can configure the access to ECU variables by configuring data access. You can configure the read/write access of the MicroAutoBox III to ECU variables.
- The ECU Interface Manager lets you configure the access to ECU calibration pages using ConfigurationDesk and the MicroAutoBox III.

The MicroAutoBox III supports the following ECU interfaces:

- DCI-GSI2
- XCP on CAN
- XCP on Ethernet

Compatibility of ECU Interface Manager 2.6

Compatibility in general

dSPACE recommends using only software products from the same dSPACE Release. This ensures maximum run-time compatibility.

Compatibility between EIC files and ConfigurationDesk

The following table shows the compatibility between EIC files and ConfigurationDesk:

	EIC Files Created With ...						
	ECU Interface Manager 2.0p ^{1), 2)}	ECU Interface Manager 2.1 ^{3), 2)}	ECU Interface Manager 2.2 ^{4), 2)}	ECU Interface Manager 2.3 ^{5), 2)}	ECU Interface Manager 2.4 ^{6), 2)}	ECU Interface Manager 2.5 ^{7), 2)}	ECU Interface Manager 2.6 ⁸⁾
ConfigurationDesk 6.4 ⁸⁾	✓	✓	✓	✓	✓	✓	✓
ConfigurationDesk 6.3 ⁷⁾	✓	✓	✓	✓	✓	✓	-
ConfigurationDesk 6.2 ⁶⁾	✓	✓	✓	✓	✓	-	-
ConfigurationDesk 6.1 ⁵⁾	✓	✓	✓	✓	-	-	-
ConfigurationDesk 6.0 ⁴⁾	✓	✓	✓	-	-	-	-
ConfigurationDesk 5.7 ³⁾	✓	✓	-	-	-	-	-
ConfigurationDesk 5.6 SP1 ¹⁾	✓	✓	-	-	-	-	-

¹⁾ dSPACE Release 2016-B

²⁾ To perform external ECU interfacing with MicroAutoBox III, the EIC file must be created with ECU Interface Manager 2.6 or later. EIC files created with ECU Interface Manager 2.5 or earlier are not supported.

³⁾ dSPACE Release 2017-A

⁴⁾ dSPACE Release 2017-B

⁵⁾ dSPACE Release 2018-A

⁶⁾ dSPACE Release 2018-B

⁷⁾ dSPACE Release 2019-A

⁸⁾ dSPACE Release 2019-B

Migrating to ECU Interface Manager 2.6

Automatic migration of projects

You can reuse projects in ECU Interface Manager 2.6 if the projects were last saved with the ECU Interface Manager 2.0 p1 or later.

When you open the projects in the ECU Interface Manager 2.6, they are migrated automatically.

Note

In the ECU Interface Manager 2.6, you cannot reuse projects that were last saved with ECU Interface Manager 2.0 or earlier.

Additional migration steps in some cases

To migrate to the ECU Interface Manager 2.6 from versions earlier than the ECU Interface Manager 2.2, you might also have to perform the migration steps of the intervening ECU Interface Manager versions.

Firmware Manager

New Features of Firmware Manager 3.0

Enhanced platform support

The Firmware Manager supports the firmware update of the following dSPACE hardware:

- MicroAutoBox III
- DS6121 Multi I/O Board
- DS6321 UART Board

Model Compare

Where to go from here

Information in this section

New Features of Model Compare 3.0	107
Migration to Model Compare 3.0	109



New Features of Model Compare 3.0

Parallel Dump

If you work with a separate MATLAB instance for each compared Simulink model, you can now specify whether to use them one after another or in parallel during model dumping. This enhances the dumping performance significantly. It is enabled by default, but you can disable it, for example, if your MATLAB configuration prohibits parallel instances.

During the dumping, the progress of the operation is displayed as separate progress bars.

Related documentation

- [How to Optimize XML Dump File Creation](#) ( [Model Compare Guide](#))
- [MATLAB Connection Page](#) ( [Model Compare Reference](#))

External Text Comparison Tool

You can now specify an external text comparison tool to display, compare, and merge properties of the comparison and the reference model. This makes it much easier to work with properties of more than three lines, such as MATLAB scripts.

You can choose from the following tools:

- Use the Show Details dialog as you did in earlier Model Compare versions.
- If *Beyond Compare*[®] is installed, use the predefined call of this tool.

- If another text comparison tool is installed, use the customized call of this tool that you specified beforehand.

Related documentation

- [How to Use an External Text Comparison Tool to Merge Scripts](#) (📖 Model Compare Guide)
- [External Text Comparison Tool Page](#) (📖 Model Compare Reference)

Working with multiple monitors

To help you to keep your desktop well-organized when you work with multiple windows, the Window Placement feature of Model Compare is enhanced. You can now define default locations for all open windows, depending on whether you compare two or three models. You can also distribute the windows across different monitors.

You can use the toolbar to activate and configure the automatic window placement.

Related documentation

- [How to Configure Default Window Positions](#) (📖 Model Compare Guide)
- [Window Placement Page](#) (📖 Model Compare Reference)
- [Window Placement](#) (📖 Model Compare Reference)

Start Page

A new Start page helps you open an existing session or create a new one. It also provides quick access to the product documentation and demos.

Related documentation

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

Improved Status Bar

Depending on the configuration of the current comparison, its state, and its result, the status bar now displays additional useful information. For example, the displayed connection states of the involved MATLAB instances let you estimate the execution time for a Auto-Highlighting Mode command.

Clicking on some of the status bar fields lets you open relevant dialogs to adjust the related comparison setting and comments.

You can expand your Model Compare working area by hiding the status bar.

Related documentation

- [How to Change the Display Options of the Tool Window, Find Bar, Legend Pane, and Status Bar](#) (📖 Model Compare Guide)
- [Status Bar](#) (📖 Model Compare Reference)
- [Show/Hide Status Bar](#) (📖 Model Compare Reference)

Additional improvements

Model Compare 3.0 also provides the following improvements:

- You can now start the dumping of an open model directly from the Model Compare menu in the MATLAB Simulink/Stateflow Model Editor. Refer

to [How to Create XML Dump Files from MATLAB](#) ([📖 Model Compare Guide](#)) and to [Dump Model](#) ([📖 Model Compare Reference](#))

- To optimize the table layout in the Model Navigator, you can now activate an automatic adjustment of the column width. This makes a maximum of relevant data visible and avoids displaying superfluous white space. Refer to [Automatic Column Fit](#) ([📖 Model Compare Reference](#)).
- The documentation on working with models that are managed via a version control system is revised. Refer to [Basics on the Interaction with Version Control Systems](#) ([📖 Model Compare Guide](#))
- The Model Compare user interface now provides an optional Legend pane in its Tool Window. This helps you interpret the detected model differences that are shown in the Model Navigator and the Property Inspector by displaying the currently used color coding. Refer to [Show/Hide Legend](#) ([📖 Model Compare Reference](#)).

Migration to Model Compare 3.0

No adaptation necessary

You can migrate from Model Compare 2.9 to Model Compare 3.0 without adaptations.

ModelDesk

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

New Features of ModelDesk 5.2	111
Migration to ModelDesk 5.2	112

New Features of ModelDesk 5.2

Platform support

ModelDesk supports MicroAutoBox III.

Parameterization

You can parameterize the ASM traffic driver. For details, refer to [New Features of ASM Traffic Blockset 4.0](#) on page 49.

Related topics**HowTos**

[How to Specify the Properties of a Traffic Driver \(📖 ModelDesk Scenario Creation\)](#)

Migration to ModelDesk 5.2

Project migration As of ModelDesk 5.2, you can migrate only projects created with ModelDesk 4.1 (Release 2015-A) and higher.

Platform support As of ModelDesk 5.1, the DS1103 PPC Controller Board is no longer supported.

Maneuver Editor As of ModelDesk 4.7, the Maneuver Editor is obsolete. You can specify maneuvers using the Scenario Editor. Maneuvers specified with the Maneuver Editor are automatically migrated to scenarios for the Scenario Editor. However, scripts that use the tool automation of the Maneuver Editor cannot be migrated. If you want to use scripts, you must activate the Maneuver Editor by using the Maneuver Compatibility command.

Note

When you enable maneuver compatibility, manual modifications of the simulation model might be required to enable proper use of the maneuver definition by the Maneuver Editor.

With Release 2019-B, the compatibility mode is provided for the last time. It will be discontinued with Release 2020-A. Therefore, it is mandatory to migrate your projects from Releases 2017-B and earlier with the current Release.

Tool automation for plotting As of ModelDesk 4.4, ModelDesk has new plotters, and the tool automation for plotting has been 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, plotting is triggered by the simulation model. Previously, ModelDesk triggered plotting. The plots are usually identical but can 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.

Related topics**References**

[Maneuver Compatibility](#) (📖 ModelDesk Scenario Creation)

Model Interface Package for Simulink

Where to go from here



Information in this section

[New Features of the Model Interface Package for Simulink 4.2](#) 113

[Migrating to the Model Interface Package for Simulink 4.2](#) 114

New Features of the Model Interface Package for Simulink 4.2

Support of typed bus signals

The Model Interface Package for Simulink now supports typed bus signals, i.e., bus signals specified by Simulink.Bus objects. You can synchronize the Simulink.Bus objects in the dialog with the Simulink.Bus objects in the MATLAB base workspace and in the model's data dictionary via the  button. Refer to [Basics of Data Port Blocks with Structured Data Ports](#) ( [Model Interface Package for Simulink - Modeling Guide](#)).

Support of Simulink inports/outports that pass nonvirtual bus signals.

The Model Interface Package for Simulink supports code generation for Simulink inports/outports at the root level of the behavior model that pass nonvirtual bus signals. Therefore, the model can provide bus signals without having to use data port blocks.

New 64 bit target system file for SIC file generation

The new dsrt64.tlc target is now available for creating SIC files. SIC files created for the dsrt64.tlc system target file can be used in VEOS Player for creating OSA files targeting Linux in future dSPACE Releases.

New API commands

The Model Interface Package for Simulink provides the following new API commands:

- `dsrt_build()` triggers DSRT code generation for a specified Simulink model. It also lets you specify to generate a Simulink implementation container for a Simulink model.

Refer to [dsrt_build](#) ([Model Interface Package for Simulink API Reference](#)).

- `dsrt_addfiles()` lets you add files to a DSRT code generation result, such as a Simulink implementation container. The `dsrt_addfiles()` API command replaces the following API commands:

- `dsrt_addfiletosic()`
- `dsrt_addlibsobjs()`
- `dsrt_addlibtosic()`
- `dsrt_addnonbuildfiles()`

Refer to [dsrt_addfiles](#) ([Model Interface Package for Simulink API Reference](#)).

Migrating to the Model Interface Package for Simulink 4.2

New API command for adding files to a code generation result

The following API commands are no longer available:

- `dsrt_addfiletosic()`
- `dsrt_addlibsobjs()`
- `dsrt_addlibtosic()`
- `dsrt_addnonbuildfiles()`

Instead, you must use the `dsrt_addfiles()` API command to add files to a DSRT code generation result, such as an SIC file. Refer to [dsrt_addfiles](#) ([Model Interface Package for Simulink API Reference](#)).

No TRC file entries for multiple labels with the same name

As of dSPACE Release 2019-B, multiple labels with the same name no longer have an entry in the TRC file by default. If required, you can temporarily activate TRC file entries for multiple labels.

Type the following commands in the MATLAB Command Window:

- To activate the option

```
ds_trc_multiplelabeloccurrence('set', 1)
```

- To reset the option to the default behavior

```
ds_trc_multiplelabeloccurrence('set', 0)
```

Note

- We strongly recommend not to activate this option.

Using multiple labels with the same name in your variable description is on your own risk.

- The option to activate TRC file entries for multiple labels will no longer be available in one of the next dSPACE Releases. It will be available only temporarily for migrating existing models or scripts that handle multiple labels.

MotionDesk

Where to go from here

Information in this section

New Features of MotionDesk 4.5	117
Migrating to MotionDesk 4.5	117

New Features of MotionDesk 4.5

Platform support

MotionDesk supports MicroAutoBox III.

Related topics

Basics

[Working with the Model and Sensor Interface Blockset \(📖 MotionDesk Calculating and Streaming Motion Data\)](#)

Migrating to MotionDesk 4.5

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, these 3-D objects are obsolete. To use an old scene, delete these objects before you activate 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 sky of the environment instead.

Migrating 3-D custom objects

If you want to use 3-D custom objects in the VRML2 format that you used in MotionDesk 2.2.1 or earlier, you have to convert the VRML2 files to 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 MotionDesk experiments in the MDX file format (used in MotionDesk 2.1.6 and earlier) or scenes stored in the 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 must migrate them 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

New Features of Real-Time Testing 4.2	119
Migrating to Real-Time Testing 4.2	119

New Features of Real-Time Testing 4.2

Platform support

MicroAutoBox III Real-Time Testing supports MicroAutoBox III with the Python 3.6.4 interpreter. To use Real-Time Testing, it must be activated in the web interface of the MicroAutoBox III.

However, Real-Time Testing does not support the following features: CAN and Ethernet communication.

VEOS Real-Time Testing supports VEOS 4.5 with the Python 3.6.4 interpreter.

Support for sleep/wake-up simulation of V-ECUs

You can use Real-Time Testing during sleep mode and the wake-up process of V-ECUs.

Migrating to Real-Time Testing 4.2

Incompatible BCG files

The BCG files generated with Real-Time Testing 4.0 or earlier cannot be used for Real-Time Testing 4.2. You must create the BCG file of the Real-Time Testing sequence again.

Only for VEOS: Because the internal Python interpreter version changed from 2.7.11 to 3.6.4, the BCG files must be generated again. You must migrate the Python sequence files from syntax of the Python 2.7.11 to the syntax of Python 3.6.4. For more information about migrating Python scripts, refer to <http://www.dspace.com/go/Python36Migration>.

C# demo scripts

As of Real-Time Testing 4.2, demo scripts in C# are not longer included. The required internal COM interface will be discontinued.

Variable access and rounding behavior

If the Simulink variable type is integer and the value written to the variable is of floating point type, the floating point value is rounded to fit the Simulink integer variable.

Related topics

Basics

[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	121
Migration Aspects of RTI/RTI-MP and RTLib	122

New Features of RTI/RTI-MP and RTLib

New features in RTI/RTI-MP

RTI and RTI-MP have the following new features:

- Supporting MATLAB R2019b
- New command to configure the handling of multiple labels with the same name

As of dSPACE Release 2019-B, multiple labels with the same name no longer have an entry in the TRC file by default. If required, you can temporarily activate TRC file entries for multiple labels.

Type the following commands in the MATLAB Command Window:

- To activate the option

```
ds_trc_multiplelabeloccurrence('set', 1)
```

- To reset the option to the default behavior

```
ds_trc_multiplelabeloccurrence('set', 0)
```

Note

- We strongly recommend not to activate this option.
Using multiple labels with the same name in your variable description is on your own risk.
- The option to activate TRC file entries for multiple labels will no longer be available in one of the next dSPACE Releases. It will be available only temporarily for migrating existing models or scripts that handle multiple labels.

Limitations when using MATLAB R2019b

The following new features introduced with MATLAB R2019b are supported by RTI/RTI-MP with limitations:

- In Simulink, you can now add a subsystem by reference. Similar to the handling of referenced models, RTI and RTI-MP support subsystem references in the model. However, using RTI blocks in a referenced subsystem is not allowed.
- Simulink Coder copies the generated code to the Simulink Cache (SLXC) file. To enforce a rebuild, you have to delete the build folder, the slprj folder, and the Simulink Cache file. If you use the `rti_build2` or `rtimp_build2` command with its `CleanUpAll` method, this is automatically done.

Migration Aspects of RTI/RTI-MP and RTLib

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 by the Simulink models when you switch to a later MATLAB version or dSPACE Release, always reset the MATLAB and Simulink preferences to their default values before you start using the models.

If you change the MATLAB version and/or the dSPACE Release version, configuration sets stored in a MAT file of an earlier version might cause problems. Therefore, you are recommended to create these configuration sets again when you change the Release version.

Note

Configuration sets stored in MAT files using dSPACE Release 2013-B or earlier cannot automatically be migrated to dSPACE Release 2019-B with MATLAB R2019b. When you load these MAT files, some settings might be lost.

Discontinued commands

The following commands are discontinued with dSPACE Release 2019-B:

- The `rti_build` and `rtimp_build` commands are no longer supported. You can use the successors `rti_build2` and `rtimp_build2` instead.

For more information, refer to [rti_build2](#) ([RTI and RTI-MP Implementation Reference](#)) and [rtimp_build2](#) ([RTI and RTI-MP Implementation Reference](#)).

End of software support for discontinued dSPACE hardware

For information on the end of software support for discontinued dSPACE hardware, refer to [Discontinuations](#) on page 17.

RTI Bypass Blockset

Where to go from here

Information in this section

New Features of the RTI Bypass Blockset 3.13	125
Migrating to RTI Bypass Blockset 3.13	125

New Features of the RTI Bypass Blockset 3.13

RTI Bypass Blockset

Support of ARM microcontrollers for on-target prototyping The RTI Bypass Blockset now supports on-target prototyping for microcontrollers of ARM microcontroller type.

RTI Bypass Blockset MATLAB API

Support of enhancements to RTI Bypass Blockset The RTI Bypass Blockset MATLAB API supports the enhancements to the RTI Bypass Blockset. Refer to the [RTI Bypass Blockset MATLAB API Reference](#).

Migrating to RTI Bypass Blockset 3.13

Working with models from earlier RTI Bypass Blockset versions 3.x and 2.x

The current Release contains RTI Bypass Blockset 3.13, 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 3.13, 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 step is performed automatically when 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 3.13 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 step is performed when 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 3.13 (.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 unchanged and accessible at the specified location.

- *Working with models from RTI Bypass Blockset 2.6 up to and including RTI Bypass Blockset 3.12*

If a Simulink model was built with RTI Bypass Blockset 2.6 up to RTI Bypass Blockset 3.12, and you open it with RTI Bypass Blockset 3.13, 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 3.12, 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 5.3	127
Migrating to RTI CAN MultiMessage Blockset 5.3	128

New Features of the RTI CAN MultiMessage Blockset 5.3

Support of static container IPDUs

The RTI CAN MultiMessage Blockset now supports container IPDUs with a static container layout.

Refer to [Aspects of Miscellaneous Supported AUTOSAR Features](#) ( RTI CAN MultiMessage Blockset Reference).

Support of cryptographic IPDUs

The RTI CAN MultiMessage Blockset now supports cryptographic IPDUs.

Refer to [Aspects of Miscellaneous Supported AUTOSAR Features](#) ( RTI CAN MultiMessage Blockset Reference).

Support of secure onboard communication for static container IPDUs

The RTI CAN MultiMessage Blockset now supports secure onboard communication (SecOC) for container IPDUs with a static container layout.

Refer to [Aspects of Miscellaneous Supported AUTOSAR Features](#) ( RTI CAN MultiMessage Blockset Reference).

Migrating to RTI CAN MultiMessage Blockset 5.3

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 a 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', bdroot)`.

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 a 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 3.8	129
Migrating to RTI FPGA Programming Blockset 3.8	131

New Features of the RTI FPGA Programming Blockset 3.8

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 ¹⁾	Operating System
Vivado 2019.1 ²⁾	<ul style="list-style-type: none"> ▪ MATLAB R2018a ▪ MATLAB R2018b ▪ MATLAB R2019a 	Windows operating system that is supported by the RCP and HIL software of the current Release. Refer to Operating System on page 220.

¹⁾ The Processor Interface sublibrary of the RTI FPGA Programming Blockset also supports MATLAB R2019b.

²⁾ The Vivado HL WebPACK Editions of the Xilinx design tools also support the DS2655 (7K160) and DS6601 FPGA base boards. A separate license for the Xilinx System Generator for DSP is required for modeling FPGA applications with the RTI FPGA Programming Blockset.


New I/O Module frameworks for MicroAutoBox III

The new *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)* and *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)* frameworks now support the DS1552/DS1552B1 Multi-I/O Module installed in the MicroAutoBox III.

- 24 x analog input channels (ADC)
- 4 x analog output channels (DAC)
- 8 x bidirectional digital channels
- 16 x digital input signals
- 16 x digital output signals
- UART interface
- 3 x digital crank/cam sensor inputs
 - To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.
- 1 x inductive zero voltage detector
 - To provide read access to an inductive zero voltage detector. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle.

The new *FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework now supports the DS1552/DS1552B1 Multi-I/O Module installed in the MicroAutoBox III.

- 5 x digital crank/cam sensor inputs
 - To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.
- 1 x inductive zero voltage detector
 - To provide read access to an inductive zero voltage detector. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle.
- 4 x knock sensor input signals
- 8 x bidirectional digital channels
- 40 x digital input signals
- 14 x analog input channels (ADC)

For more information on the DS1552 frameworks, refer to [RTI Block Settings for the FPGA1403Tp1 with Multi-I/O Module Frameworks](#) ( RTI FPGA Programming Blockset - FPGA Interface Reference).

For more information on the DS1554 framework, refer to [RTI Block Settings for the FPGA1403Tp1 with Engine Control I/O Module Framework](#) ( RTI FPGA Programming Blockset - FPGA Interface Reference).

Enhancements to the SCALEXIO FPGA base board frameworks

The *DS6601 (KU035) FPGA Base Board* provides the following enhancements:

- Support of Samtec FireFly™ ECUO-B04 (QSFP+) multi-gigabit transceiver (MGT) to establish a communication between two FPGA boards, for example. The frameworks supports the Aurora 64B/66B framing protocol.

The *DS6602 (KU15P) FPGA Base Board* provides the following enhancements:

- Support of Samtec FireFly™ ECUO-B04 (QSFP+) multi-gigabit transceiver (MGT) to establish a communication between two FPGA boards, for example. The framework supports the Aurora 64B/66B framing protocol.
- Support of the onboard 4 GB DDR4 RAM.

For more information on the framework for the MGT interface, refer to [RTI Block Settings for the DS660X_MGT Framework](#) ([RTI FPGA Programming Blockset - FPGA Interface Reference](#)).

For more information on the RAM support, refer to [Accessing the DDR4 RAM of the DS6602](#) ([RTI FPGA Programming Blockset Guide](#)).

General enhancements

Support of Simulink buses The SCALEXIO and the MicroAutoBox III frameworks now support Simulink buses to implement the communication between the real-time processor and the FPGA application.

For more information, refer to [How to Use Simulink Buses for Modeling the Processor Communication](#) ([RTI FPGA Programming Blockset Guide](#)).

Related topics

Basics

[Migrating to RTI FPGA Programming Blockset 3.8](#) 131

Migrating to RTI FPGA Programming Blockset 3.8

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 3.8


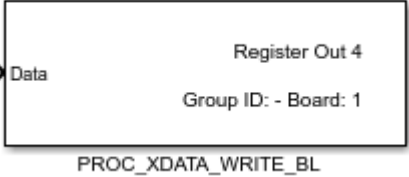
If you implemented an FPGA application with RTI FPGA Programming Blockset Version 1.1 and later and want to use it with RTI FPGA Programming Blockset 3.8, the framework automatically updates itself to the current framework version.

The update affects all the subsystems in the model/subsystem. The parameters of the blocks stay the same after updating to the current framework version.

Display 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 sublibrary. The migration process does not change the size of the original blocks to keep the block arrangement of the model. Therefore, the display of the

migrated blocks is different from the default display of processor interface blocks. The following illustrations provide an example.

Display After Migration	Default Display
	

ConfigurationDesk custom functions incompatible with current dSPACE Release

FPGA custom function block types that are not built with the RTI FPGA Programming Blockset 3.8 might be incompatible with the current ConfigurationDesk version.

FPGA Programming Blockset 3.5 or earlier As of dSPACE Release 2018-B, the angle range handling of the angular processing unit (APU) was changed. FPGA custom function blocks that use the APU in the 360° angle range are incompatible if they are built with the FPGA Programming Blockset 3.5 or earlier. To resolve the incompatibility, use the FPGA model/code of the incompatible FPGA custom function block and build a new FPGA custom function block with the RTI FPGA Programming Blockset 3.6 or later. The RTI FPGA Programming Blockset automatically migrates the framework of the FPGA model/code to the current version.

RTI FPGA Programming Blockset 2.5 An FPGA custom function block generated with RTI FPGA Programming Blockset 2.5 from dSPACE Release 2013-A and the real-time applications 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 Different FPGA Hardware](#) ([RTI FPGA Programming Blockset Guide](#)).

RTI LIN MultiMessage Blockset

Migrating to RTI LIN MultiMessage Blockset 3.3

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 a 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', bdroot)`.
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](#)).

RTI Synchronized Time Base Manager Blockset

New Features of the RTI Synchronized Time Base Manager Blockset 1.3

Support of MicroAutoBox III

The RTI Synchronized Time Base Manager Blockset now also supports the new MicroAutoBox III.

SCALEXIO Firmware

New Features of the SCALEXIO Firmware 4.5

New supported hardware

The SCALEXIO firmware supports the following new I/O boards:

- DS6121 Multi-I/O Board

The board provides features for developing electric motor control applications:

- Processing of position sensors:

- Hall sensors
- Digital incremental encoders
- Sine encoders
- Protocol-based encoders (EnDat 2.1 and 2.2, SSI, BiSS-C)
- Resolver

- Generation of multi-channel PWMs for block or sine commutation

- Phase current measurement for up to 6 phases with triggered conversion start

- +5 V onboard power supply voltage for connected sensors

- DS6321 UART Board

The board provides four independent UART channels each supporting RS232, RS422, RS485, or K-Line at a time.

DS6001 Processor Board

The firmware of the DS6001 Processor Board has a new feature:

DS2907 support The DS6001 supports the DS2907 Battery Simulation Controller. The DS2907 is used to control the current and voltage values of battery simulation in a SCALEXIO system. The controller provides two sockets for battery simulation modules to control up to two independent battery simulation power supply units.

Non-volatile data The DS6001 supports non-volatile data which is kept after the DS6001 is powered down. You can use non-volatile data, for example, to store the mileage or driver-specific settings, such as the seat position.

Related topics

Basics

Discontinuations	17
DS2907 Battery Simulation Controller (📖 SCALEXIO Hardware Installation and Configuration)	
DS6121 Multi-I/O Board (📖 SCALEXIO Hardware Installation and Configuration)	
DS6321 UART Board (📖 SCALEXIO Hardware Installation and Configuration)	

SYNECT

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

New Features of SYNECT 2.8	140
Migrating to SYNECT 2.8	145

New Features of SYNECT 2.8

Where to go from here

Information in this section

New General Features of SYNECT	140
Provides an overview of new general SYNECT features.	
New Features of Test Management	141
Provides an overview of the new SYNECT test management features.	
New Features of Workflow Management	143
Provides an overview of the new SYNECT workflow management features.	

New General Features of SYNECT

Improvements of the SYNECT server

Technical improvements The implementation of the SYNECT server now uses ASP.NET Core. This provides improved flexibility of supported technologies and platforms.

Note

Operating productive servers in Microsoft IIS

You now have to operate productive SYNECT servers in Microsoft Internet Information Services (IIS).

This is because using servers as a Windows Service now supports only HTTP communication that is not encrypted. It is recommended to use Windows Services only for development servers.

Changed installation The SYNECT server and license server installation has been changed.

You can now install the SYNECT server and the license server separately. The installation is optimized for server machines and requires minimum disk space. However, the server user documentation is now installed with the SYNECT client.

- The SYNECT server installation provides the SYNECT Server Administrator and the SYNECT Database Migrator.
- The SYNECT license server installation provides the SYNECT License Server Administrator.

Improvements for linking items

The following improvements for linking items were made.

- Linking items has been improved. The Link dialog now provides information on source and target item types of link types. This helps you if multiple link types are available even if they have the same name.
- Test and workflow management now provide the Links pane that lets you manage links of test and workflow management items.

The columns of test management grids that displayed links have been removed. You can now use the Links pane.

Discontinuation of the Script Sequencer

As of dSPACE Release 2020-A the Script Sequencer for executing Python scripts in a sequence will no longer be available.

However, the alternative will be to execute the Python scripts with workflow management.

New Features of Test Management

Introduction

SYNECT provides evaluations for separating test execution and test result evaluation.

Basics on evaluating test results

You can use evaluations for the following purposes:

- To save time when executing tests on HIL simulators.

Evaluating tests in a later step, e.g., on a PC can help you reduce the execution time of tests on simulators.

- To use test results for multiple evaluations.

Separating test execution from result evaluation lets you use test result data multiple times, e.g., to rework evaluation functions or use different evaluation functions.

Test results and evaluations Test automation tools such as AutomationDesk let you implement test cases and result evaluations separately. Test case execution returns test case results, such as capture data, MDF files, and LOG files that you can evaluate to generate a verdict.

Improved SYNECT support for evaluating test results

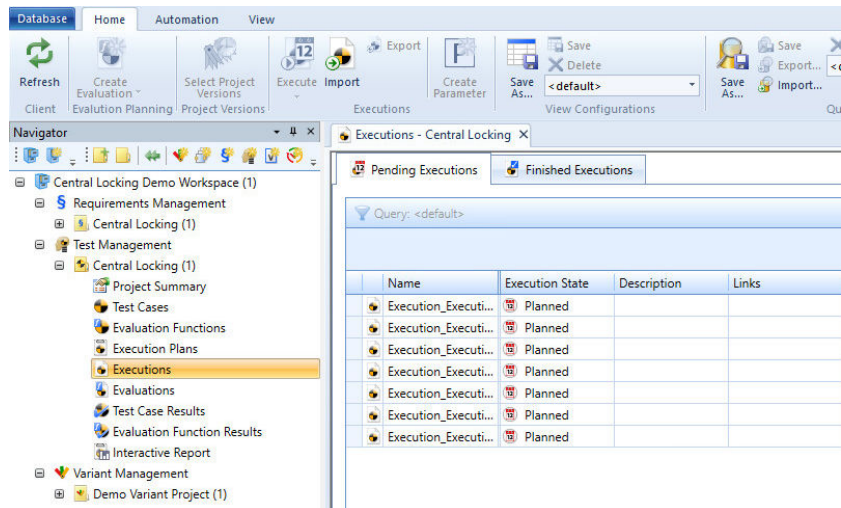
With this version the SYNECT support for evaluations has been improved in the following points:

Changed data grids Test management now provides a set of data grids that has been changed to better support evaluations.

You can now access test management items in the following data grids:

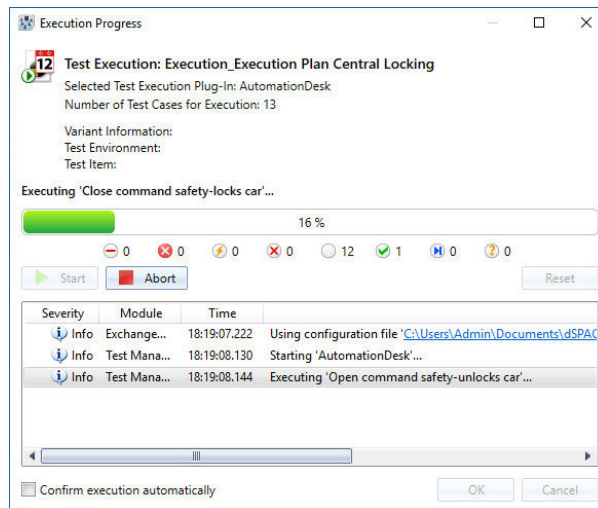
- Test Cases
- Evaluation Functions

- Execution Plans
- Executions with the following tabbed pages: Pending Executions and Finished Executions
- Evaluations with the following tabbed pages: Pending Evaluations and Finished Evaluations
- Test Case Results
- Evaluation Function Results



Changed Execution Progress dialog The Execution Progress dialog has been changed to support evaluations.

Execution of test cases or evaluation functions now starts immediately and progress messages are written to the dialog. However, the dialog no longer displays test case results and the order in which they are executed. If required, you can use a query to get this information if executions or evaluations were prepared.



Evaluation parameters You can now add parameters to evaluations.

AutomationDesk support The AutomationDesk plug-in now supports evaluation functions, evaluation function results, and evaluations.

This lets you execute evaluation functions you implemented in AutomationDesk and manage data with SYNECT.

The AutomationDesk plug-in is now implemented as a Python plug-in. You have to migrate ECXML files you used with prior versions of the AutomationDesk plug-in.

Further reading

Refer to [Evaluating Test Results](#) ([SYNECT Guide](#)).

New Features of Workflow Management

Using workflows in workflows

You can now use workflows recursively.

This has the following benefits:

- It simplifies the development of verified workflows that perform common tasks.

Sequences of workflow steps are often used to perform common tasks such as the preparation of a MATLAB model by executing workflow steps to start MATLAB, load workspace variables, load a Simulink model, and configure a model variant. You can now create and test workflows that prepare common tasks to develop a set of verified workflows.

- It simplifies the creation of complex workflows.

If you want to prepare a HIL system for testing, you often have to prepare your tests using different tools such as ControlDesk, ModelDesk, etc. You can now go back to verified workflows that perform a complete preparation task in a tool such as starting ControlDesk, loading a project, downloading an

application to the HIL system, loading layouts and parameters without the need to add the required tasks step by step. You can now focus on unique workflow steps.

Creating workflows You can use the Step Explorer to add steps to a workflow. The Step Explorer now additionally provides workflows.

Providing interface parameters You can define an interface of workflow steps that lets you provide argument values for step execution.

Workflow management has the following options to provide argument values:

- On the Interface page of workflow steps. The provided arguments are used as default values of the workflow step instances in the context of workflows.
- On the Interface page of workflows. You can select each workflow step and override the default value that was specified in the step definition.

If you use workflows recursively the values are inherited, i.e., the values of the steps of a workflow become the default values of the steps in the context of parent workflows.

- You can override interface argument values of workflow steps if you execute workflows interactively using the Start Workflow wizard.

Starting workflows The workflow management toolbox provides the WFM Starter and the WFM Job Starter to start workflows. From SYNECT, you can start workflows interactively using the Start Workflow wizard that lets you override interface argument values and deactivate workflow steps.

When you start workflows the contained steps are executed in a sequence.

Further reading

Refer to [Managing Workflows](#) ( SYNECT Guide).

Migrating to SYNECT 2.8

Where to go from here

Information in this section

Migrating Databases	145
To use the data from previous SYNECT versions with SYNECT 2.8, you have to migrate the SYNECT database.	
Migrating from SYNECT 2.7	146
You have to migrate scripts and ECXML files from SYNECT 2.7 to SYNECT 2.8.	
Data Model Changes From SYNECT 2.7 to SYNECT 2.8	149
Some parts of the SYNECT data model have been changed from SYNECT 2.7 to SYNECT 2.8.	

Migrating Databases

Introduction

To use the data from previous SYNECT versions with SYNECT 2.8, you have to migrate the SYNECT database.

To migrate databases for SYNECT Versions 2.0 - 2.7 to SYNECT 2.8, SYNECT 2.8 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](#) ([PDF](#) [The SYNECT Server Guide](#)).

Discontinuation of SQL Server versions

As of SYNECT 2.8, SQL Server 2008 and SQL Server 2008R2 versions are no longer supported.

Discontinuation of operating system versions on the SYNECT server

dSPACE support of Windows Server 2008 R2 will end with dSPACE Release 2019-B (November 2019). Microsoft® is planning to end its support for Windows Server 2008 R2. The extended support will end on the January 14, 2020. Thereafter, Microsoft will no longer provide security patches and new support information. Therefore, dSPACE Release 2019-B will be the final software version that will be released for Windows Server 2008 R2.

Migrating from SYNECT 2.7

Migrating Python plug-ins

You can write Python plug-ins to customize data exchange. With SYNECT 2.8, you have to migrate self-written Python plug-ins.

Python plug-ins now provide a class for variant references, which lets you access the following additional information:

- Name of the variation point
- Hierarchy of the variant starting from the variation point

You have to use the new `VariantReference` class in Python plug-in scripts if you collect variant references using the following properties:

- `VariantConfiguration.VariantReferences`
- `VariantDependency.References`
- `Execution.VariantReferences`
- `Execution.TestEnvironmentVariantReferences`
- `Execution.TestItemVariantReferences`
- `TestCaseResult.VariantReferences`

Old listing The following old listing uses the `Reference` class for referencing variants, which is no longer supported.

```
### The Variant Configuration provides a List for variant references and a
### Variant Model Subset reference.
### Create a reference with the API and append it to the List/item

variantConfiguration = VariantHandling.VariantConfiguration()
variantConfiguration.Name = "DE-V8"
variantConfiguration.ForeignId = "DE-V8"

# The Variant Model Subset reference
reference_vmss = Base.Reference()
reference_vmss.Name = "Country-Engine"
reference_vmss.ForeignId = "Country-Engine"
variantConfiguration.VariantModelSubsetReference = reference_vmss

# The Variant references
reference_vc1 = Base.Reference()
reference_vc1.Name = "DE"
reference_vc1.ForeignId = "DE"
variantConfiguration.VariantReferences.Add(reference_vc1)

reference_vc2 = Base.Reference()
reference_vc2.Name = "V8"
reference_vc2.ForeignId = "V8"
variantConfiguration.VariantReferences.Add(reference_vc2)

variantConfigurations.Add(variantConfiguration)
```

New listing You now have to use the VariantReference class as shown in the following listing:

```

### The Variant Configuration provides a List for variant references and a
### Variant Model Subset reference.
### Create a reference with the API and append it to the List/item

variantConfiguration = VariantHandling.VariantConfiguration()
variantConfiguration.Name = "DE-V8"
variantConfiguration.ForeignId = "DE-V8"

# The Variant Model Subset reference
reference_vmss = Base.Reference()
reference_vmss.Name = "Country-Engine"
reference_vmss.ForeignId = "Country-Engine"
variantConfiguration.VariantModelSubsetReference = reference_vmss

# The Variant references
reference_vc1 = Base.VariantReference()
reference_vc1.Name = "DE"
reference_vc1.ForeignId = "DE"
variantConfiguration.VariantReferences.Add(reference_vc1)

reference_vc2 = Base.VariantReference()
reference_vc2.Name = "V8"
reference_vc2.ForeignId = "V8"
variantConfiguration.VariantReferences.Add(reference_vc2)

variantConfigurations.Add(variantConfiguration)

```

Migrating the ECXML files for the AutomationDesk plug-in

You can configure plug-ins for SYNECT via ECXML files. With SYNECT 2.8, you have to split ECXML files you used with previous versions for the AutomationDesk plug-in into two files that deal separately with the *import* of test cases and with their *execution*. You also have to add the following properties:

ImportAdapter You have to add this property to the ECXML file that deals with the import of test cases. The property defines the path to the Python adapter that handles the import of test cases:

```

<Property Key="ImportAdapter"
Value="..\Source\Client\AUDImportAdapter.pyc" />

```

ExecuteAdapter You have to add this property to the ECXML file that deals with the execution of test cases. The property defines the path to the Python adapter that handles the execution of test cases:

```

<Property Key="ExecuteAdapter"
Value="..\Source\Client\AUDExecuteAdapter.pyc" />

```

You have to make adjustments to the following properties:

InitSequences With SYNECT 2.8, lists of initialization sequences are not supported anymore. You must change the property key to `InitSequenceName`. If there is more than one initialization sequence, you can specify a folder which contains the sequences.

With SYNECT 2.8, the AutomationDesk plug-in supports evaluations and evaluation function results. As a consequence, you must change the following property keys:

Old name	New name
TestCaseName	ImportItemName
InputExecutionInformationDict	InputExecuteItemName
InputTestCaseResultInformationDict	InputResultItemInformationDict
OutputExecutionInformationDict	OutputExecuteItemInformationDict
OutputTestCaseResultInformationDict	OutputResultItemInformationDict

The ProjectLink property is not supported anymore. If you need the corresponding functionality, please contact dSPACE support.

Further information For more information, refer to ECXML properties of the AutomationDesk plug-in (refer to AutomationDesk ([SYNECT Guide](#))).

Migrating client API scripts

The client API lets you get information on the active view of SYNECT.

With this version, the Pending Executions and Finished Executions views were merged to the Executions view. You have to adapt client API scripts accordingly.

Old listing The following old listing shows how you were able to test if the Pending Executions or Finished Executions views were active. With this version, the listing does not result in an error. However, the views are not used by SYNECT any longer.

```
import dspace.com
Enums = dspace.com.Enums(Application)

# View from Application object or View(De)Activating event
view = Application.ActiveView

if view.Type == Enums.ViewType.PendingExecutions:
    Application.Log.WriteInformation(">>> Pending Executions View")
elif view.Type == Enums.ViewType.FinishedExecutions:
    Application.Log.WriteInformation(">>> Finished Executions View")
```

New listing The following new listing shows how to test if the tabbed pages that were introduced with this version are active.

```
import dspace.com
Enums = dspace.com.Enums(Application)

# View from Application object or View(De)Activating event
view = Application.ActiveView

if view.Type == Enums.ViewType.Executions:
    if view.SelectedTab == Enums.ExecutionsTab.PendingExecutions:
        Application.Log.WriteInformation(">>> Pending Executions View")
    elif view.SelectedTab == Enums.ExecutionsTab.FinishedExecutions:
        Application.Log.WriteInformation(">>> Finished Executions View")
```

Data Model Changes From SYNECT 2.7 to SYNECT 2.8

Introduction Some parts of the SYNECT data model have been changed from SYNECT 2.7 to SYNECT 2.8.

Deleted item types Item types were not deleted.

Deleted attributes Attributes were not deleted.

Deleted reference types Reference types were not deleted.

New item types Item types were not added.

New attributes Attributes were not added.

New reference types The following reference types have been added to the SYNECT data model:

Name	Source	Target
Parameters/Evaluation Function	Evaluation Function (SYNECT Data Model Reference)	Parameter (SYNECT Data Model Reference)
Parameters/Evaluation	Evaluation (SYNECT Data Model Reference)	Parameter (SYNECT Data Model Reference)
Parameters/Evaluation Function Result	Evaluation Function Result (SYNECT Data Model Reference)	Parameter (SYNECT Data Model Reference)
Overwrite Path/Overwritten By	Interface Argument Value (SYNECT Data Model Reference)	Workflow Step Instance (SYNECT Data Model Reference)

SystemDesk

Where to go from here

Information in this section

New Features of SystemDesk 5.4	152
Migrating to SystemDesk 5.4	156

New Features of SystemDesk 5.4

Where to go from here

Information in this section

New General Features	152
Provides information on new general features.	
Splittable AUTOSAR Elements	152
SystemDesk now supports the splittable mechanism by assigning splittable AUTOSAR elements across several master files.	
Configuring ECUs	154
Provides information on new features for configuring ECUs.	

New General Features

Classic Platform support by SystemDesk 5.4

AUTOSAR release for modeling SystemDesk lets you model Classic Platform software and system architectures with a data model according to the AUTOSAR CP 4.4.0, AP 19-03 Release. However, SystemDesk also lets you exchange data according to other AUTOSAR releases.

Data exchange support SystemDesk supports AUTOSAR CP 4.4.0, 4.3.1, 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.

Adaptive Platform support

SystemDesk now supports AUTOSAR AP 19-03 for developing Adaptive Platform software. The revision is also supported for exchanging data.

Splittable AUTOSAR Elements

Support for distributing AUTOSAR elements

AUTOSAR has defined the splittable mechanism to distribute AUTOSAR elements across several AUTOSAR files. This supports workflows where OEMs and suppliers develop AUTOSAR elements in collaboration projects.

SystemDesk now supports the splittable mechanism by assigning splittable AUTOSAR elements across several master files.

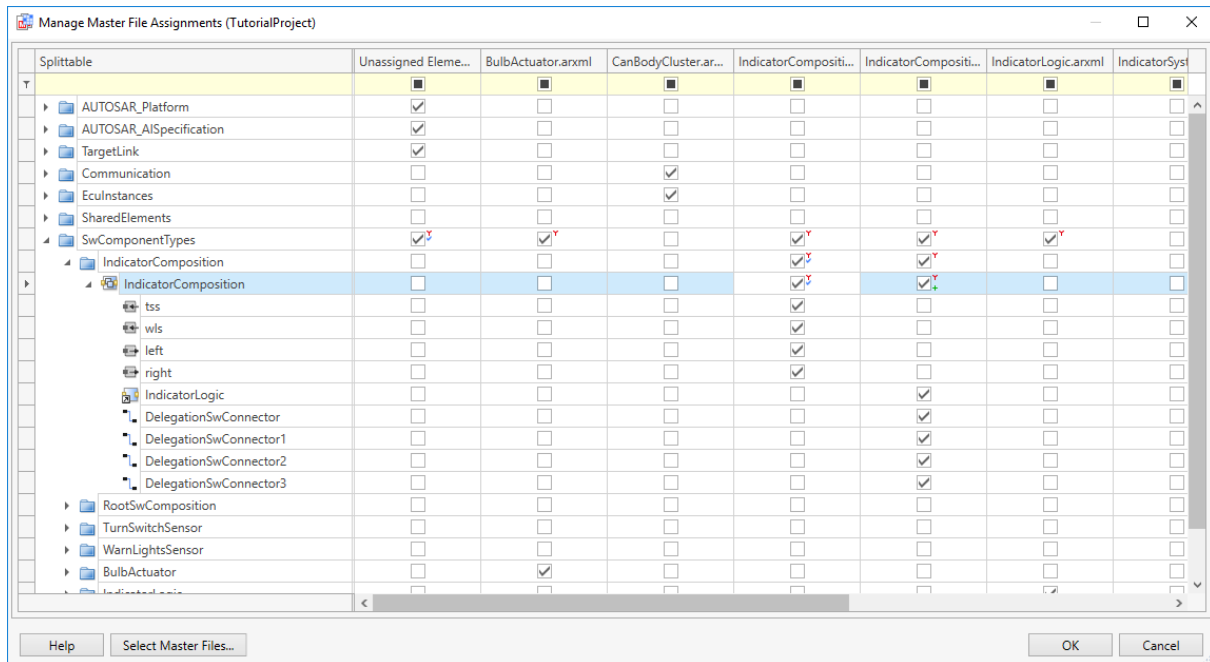
The following terms are essential for the splittable mechanism.

Split Generic term for the parts that in combination define an element.

Main split According to AUTOSAR, only one split, i.e., the main split carries the information that cannot be split, such as properties of the root node of the AUTOSAR element.

Tutorial example

The following illustration shows an example for distributing AUTOSAR elements across master files. This example is included in the SystemDesk Tutorial.



This is an example of an OEM-supplier-workflow. The OEM provides the interface of a composition that is to be implemented by the supplier. The supplier in turn implements software component types with the required functionality. To avoid changes of the OEM definition and to be prepared for updates of the composition interface by the OEM, the supplier specifies the internal structure of the composition in a separate file.

The following table lists the assignments of AUTOSAR elements to master files in SystemDesk:


Master File	Project Party	Contents
IndicatorComposition.arxml	OEM	Composition interface, i.e., composition type and ports
IndicatorComposition_Internal.arxml	Supplier	Internal composition structure, i.e., contained component prototypes and delegation connectors
IndicatorLogic.arxml	Supplier	Atomic component type

Managing master file assignments

The Master File Explorer lets you manage master files and the assignment of AUTOSAR elements to the files. The explorer now lets you additionally manage the assignment of splittables to master files. SystemDesk also provides a dedicated dialog for this purpose.

The support for distributing AUTOSAR elements across master files by file assignments provides the following options:



Splitting elements across files You can add splits of elements to the available master files.

Elements that are split across files are indicated by the following symbol: 

Specifying the main split You can specify the main split of a set of splits.

Elements that represent the main split are indicated by the following symbol: 

Adding child elements of a splittable You can specify the master file to which child elements are added when you create elements with SystemDesk.

Elements to which child elements of a splittable element are added are indicated by the following symbol: . If child elements are added to the file with the main split, this is indicated by the following symbol: 

Moving splits You can move splits to other master files. SystemDesk lets you select from the available master files for this.

Merging splits You can merge all the splits of an element in one master file. You have to select the split in the file where SystemDesk is to merge the element.

Further reading

For more details, refer to [Basics on Assigning Elements](#) ( [SystemDesk Manual](#)).

Configuring ECUs

Additional support for generating virtual ECUs

SystemDesk provides basic software modules of the microcontroller abstraction layer (MCAL) for the integration of third-party basic software in a virtual ECU. With this version, SystemDesk provides an additional MCAL module and dSPACE-specific modules for emulating ECU hardware.

The following additional modules are provided:

- Serial Peripheral Interface Handler/Driver (Spi)
- Angular processing unit (Apu)
- Edge driver (Edge)

Serial Peripheral Interface Handler/Driver (Spi)

Module of the microcontroller abstraction layer (MCAL) for communication with devices that are connected via SPI buses or onchip SPI devices.

The module lets you simulate buffering and data transmission.

Angular processing unit (Apu) dSPACE-specific module to emulate hardware that measures specific angles of rotating shafts.

You can use the Apu module if you connect a V-ECU to an environment model that simulates, e.g., a crank shaft. The module lets you create events for specific angles. You can reference channels of an ICU or callback functions of complex device drivers to transmit the events.

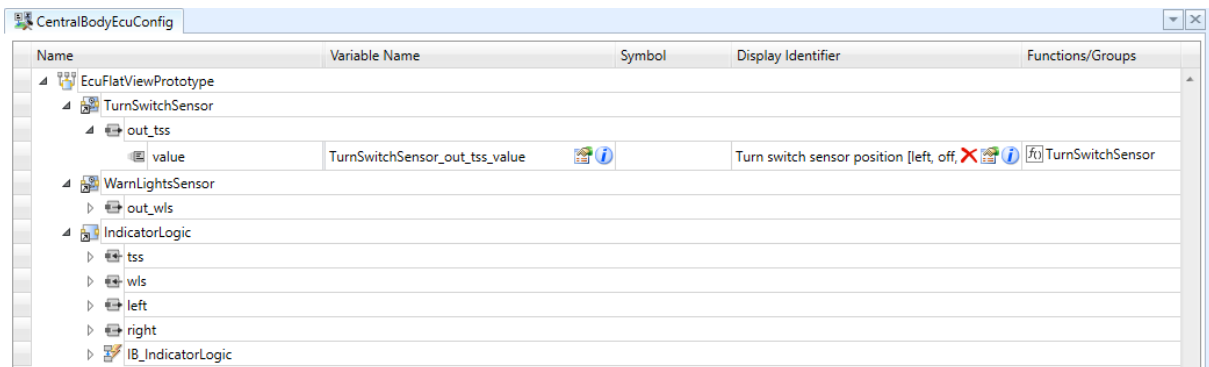
Edge driver (Edge) dSPACE-specific module to emulate ECU hardware that detects signal edges.

You can use the Edge module if you connect a V-ECU to an environment model that simulates, e.g., an ignition switch. The module lets you detect falling and rising signal edges. You can reference ICU channels or specify callback functions of complex device drivers to route signal edges to basic software.

Sleep/Wakeup support The simulator abstraction now provides an API function that lets you simulate scenarios where the ECU state manager switches an ECU to a *sleep* state. *Wakeup* can be performed via a polling mechanism of the ECU state manager or interrupt events that are triggered by BSW modules.

Editing measurement and calibration data of V-ECU variables The implementation of a virtual ECU contains a variable description of ECU variables that is created during RTE generation.

SystemDesk now provides a Variable Editor that lets you edit the measurement and calibration data of ECU variables.



Further reading Refer to [Configuring ECUs](#) ([SystemDesk Manual](#)).

Migrating to SystemDesk 5.4

Migrating to SystemDesk 5.4

Automatic migration of projects

SystemDesk 5.4 automatically migrates SystemDesk 5.2, and 5.3 SDP project files when it loads.

Note

You are recommended to install the most recent patch for SystemDesk 5.2 or 5.3. Then, save the SDP project files you want to migrate before opening them in SystemDesk 5.4.

Migrating from SystemDesk 5.3

Migrating scripts for automating SystemDesk The SystemDesk API was changed as of SystemDesk 5.4. Some interfaces were added compared to SystemDesk 5.3 and certain interfaces were changed. For more information, refer to *API Changes from SystemDesk 5.3 to SystemDesk 5.4* in the *SystemDesk API Reference* that is only available in dSPACE Help.


TargetLink

Where to go from here

Information in this section

New Features of TargetLink 5.0	158
Migrating to TargetLink 5.0 and TargetLink Data Dictionary 5.0	175
Changes in Future TargetLink Versions	211

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 5.0

Where to go from here

Information in this section

Modeling in Simulink or Stateflow	158
MATLAB® Code	160
Adaptive AUTOSAR	161
Classic AUTOSAR	162
Tool Chain Support	164
Code Generation Core Functionality	164
Target Simulation (PIL)	166
Usability	167
Code Generator Options	170
API Functions and Hook Scripts	171
Other	173

Modeling in Simulink or Stateflow

Where to go from here

Information in this section

Improved Support of Array-of-Structs	158
Data Stores Usable Across Code Generation Units	159
N-Dimensional Look-Up Tables	159
Other MATLAB®/Simulink/Stateflow Features	159

Improved Support of Array-of-Structs

Element-wise access to array-of-structs variables via Data Store Read/Write blocks

TargetLink Data Store Memory blocks can reference DD Variable objects holding array-of-structs C variables.

Element-wise access to array-of-structs C variables can be specified in the model via TargetLink Data Store Read and Data Store Write blocks.

Related documentation


- [Working with Array-of-Bus Signals](#) ( [TargetLink Preparation and Simulation Guide](#))

Data Stores Usable Across Code Generation Units

Using Data Stores across CGUs

TargetLink can now consider data stores during code generation even though they are specified outside the incremental subsystem or referenced model for which code is being generated.

This is possible by specifying global data stores in the Data Dictionary via DD Block objects whose BlockType property is set to TL_DataStoreMemory. TargetLink Data Store Memory blocks can reference these objects and Simulink.Signal objects can be associated with these objects.

Related documentation [Working with Array-of-Bus Signals](#) ( [TargetLink Preparation and Simulation Guide](#))

N-Dimensional Look-Up Tables

Using n-dimensional Look-Up tables for DD-based code generation

Via Interpolation DD Look-Up table objects you can centrally specify look-up tables with up to five dimensions in the Data Dictionary. You can use these objects for DD-based code generation.

Using them in DD-based code generation lets you generate Look-Up table arrays with n dimensions in code modules that are independent of model elements.

You can use these generated arrays in combination with external or legacy implementations of n-dimensional table functions for three or more dimensions.

Related documentation

- [How to Specify an N-Dimensional DD Look-Up Table Object for Code Generation](#) ( [TargetLink Preparation and Simulation Guide](#))

Other MATLAB®/Simulink/Stateflow Features

Bitwise not for numerical values

For MATLAB® code and Stateflow® action language, TargetLink 5.0 additionally supports bitwise not-operations applied directly to numerical values, if the

operation is the last operation on the right-hand side of an assignment. Additionally, all of the following conditions for the left-hand side of the assignment must be fulfilled:

- The left-hand side is not scaled.
- The left-hand side is not saturated.
- The data type of the left-hand side is not a floating-point type.

Related documentation

- None

Stateflow chart blocks available in library

With TargetLink 5.0, Stateflow® Chart blocks are available in the TargetLink block library for supported Simulink blocks. The Action Language property of these blocks has been pre-set to C.

Related documentation

- [Code-Relevant Simulink Blocks](#) ([📖 TargetLink Model Element Reference](#))

SPI blocks without preceding blocks

For SPI blocks that are not connected to preceding blocks or that are connected to a Ground block, the block output inherits the data type `Int32` and the default `VOID_SCALING` scaling. TargetLink displays W15295.

Related documentation

- W15295

MATLAB® Code

MATLAB® Code Improvements

MATLAB code struct support

TargetLink 5.0 supports the Simulink.Bus data type for MATLAB variables except local MATLAB variables. As with buses in Stateflow, these variables are represented by structured types in the production code.

Related documentation

- [Basics on Supporting MATLAB® Code Using TargetLink](#) ([📖 TargetLink Code Generation Guide for MATLAB® Code in Simulink® Models](#))
- [Details on MATLAB® Functions](#) ([📖 TargetLink Code Generation Guide for MATLAB® Code in Simulink® Models](#))

Comments in MATLAB code

TargetLink 5.0 lets you transfer code comments from your MATLAB® code to the generated production code. This improves the readability and traceability of the generated production code.

Related documentation

- [Basics On Transferring Comments From MATLAB Code to Production Code](#) ([📖 TargetLink Code Generation Guide for MATLAB® Code in Simulink® Models](#))

Automatic use of Simulink data types in MATLAB code functions

For any local MATLAB variables you did not specify explicitly as Data Dictionary objects, TargetLink 5.0 provides new Code Generator options that let you use the specified Simulink® data type as the basis for code generation. This increases the efficiency of production code that was generated with default settings.

Related documentation

- [Details on Determination and Mapping of Simulink® Data Types to TargetLink Data Types](#) ([📖 TargetLink Code Generation Guide for MATLAB® Code in Simulink® Models](#))
- [New Code Generator Options](#) on page 170

Adaptive AUTOSAR

Adaptive AUTOSAR

Support of ara::com

AUTOSAR Adaptive Platform, also called Adaptive AUTOSAR, is a standard based on a service-oriented architecture that aims at on-demand software updates and high-end functionalities.

TargetLink 5.0 supports select features of AUTOSAR Adaptive Platform Release 19-03 . The modeling constructs and Data Dictionary workflow for Adaptive AUTOSAR closely resemble the ones used for Classic AUTOSAR.

TargetLink 5.0 supports the Adaptive AUTOSAR standard as follows:

- Import and export of Adaptive AUTOSAR ARXMLs
- Modeling of selected parts of a service-based communication as described by ara::com:
 - Behavior of methods on server side via subsystems and the TargetLink Function block
 - Method calls on client side via subsystems and the TargetLink Function block
 - Access to fields of service interfaces
 - Configurable service discovery
- Generation of C/C++ code to use in adaptive applications

Related documentation

- *Adaptive AUTOSAR-compliant models with TargetLink:*
http://www.dspace.com/go/tl_aar

Classic AUTOSAR

Where to go from here

Information in this section

Supported AUTOSAR Releases	162
Improved Support of Data Store Blocks for AUTOSAR Communication	163
Improved Import and Export for AUTOSAR 4.4	163

Supported AUTOSAR Releases

Supported AUTOSAR Releases

The following AUTOSAR Releases are supported:

AUTOSAR Release	Revision
4.4	4.4.0 ¹⁾
4.3	4.3.1 4.3.0
4.2	4.2.2 4.2.1
4.1	4.1.3 4.1.2 4.1.1
4.0	4.0.3 4.0.2

¹⁾ New in TargetLink 5.0

AUTOSAR 2.x/3.x no longer supported TargetLink no longer supports code generation for AUTOSAR 2.x/3.x. For more information, refer to [AUTOSAR 2.x/3.x no Longer Supported](#) on page 187.

Improved Support of Data Store Blocks for AUTOSAR Communication

AUTOSAR communication with array-of-struct data types is now possible via data store blocks for the following communication kinds:

- Interrunnable communication
- Sender-receiver communication
- NvData communication

Related documentation

- [Working with Array-of-Bus Signals](#) ( [TargetLink Preparation and Simulation Guide](#))

Improved Import and Export for AUTOSAR 4.4

IsOptional element for struct components

TargetLink can now import and export the `isOptional` attribute of `APPLICATION-RECORD-ELEMENT` and `IMPLEMENTATION-DATA-TYPE-ELEMENT` elements. This attribute provides information on whether the `APPLICATION-RECORD-ELEMENT` or `IMPLEMENTATION-DATA-TYPE-ELEMENT` is available at runtime.

In the Data Dictionary, this attribute is represented by the `IsOptional` property of `DD ApplicationDataTypeComponent` and `TypedefComponent` objects. The property can be specified if these objects specify a primitive data type or an array data type.

DisplayPresentation element

TargetLink can import and export the `DisplayPresentation` attribute of the `SW-DATA-DEF-PROPS`. This attribute controls the presentation of the related data in a measurement and calibration tool.

In the Data Dictionary, this attribute is represented by the `DisplayPresentation` property of `DD ApplicationDataTypeComponent`, `TypedefComponent` objects. The property can be specified if these objects specify a primitive data type or an array data type.

Offset element

TargetLink can import and export the `offset` attribute of `TIMING-EVENT` elements.

In the Data Dictionary, this attribute is represented by the `Offset` property of `DD RteEvent` objects. The property can be specified if these objects specify a timing event.

Tool Chain Support

Using Several TargetLink-Generated SIC Files in the Same Application Process

You can now assign TargetLink-generated SIC files together with Simulink models, bus simulation container files and other SICs to the same application process in ConfigurationDesk.

Related documentation

- [Assigning multiple model implementations to the same application process in ConfigurationDesk](#) (📖 [TargetLink Interoperation and Exchange Guide](#))

Code Generation Core Functionality

Where to go from here

Information in this section

Support of Modern Standards of C	164
Void casts in function calls	165
Generating Code for 64-Bit Linux Targets	165
Improved Code Efficiency	165

Support of Modern Standards of C

Support of C99/C11

TargetLink 5.0 supports the common subset of C90 and C99/C11. It is now possible to generate production code that fully complies with the C99 and C11 standard. Additionally, select features of C99 are supported.

TargetLink 5.0 supports the specification of custom restrictions for identifiers. This lets you prevent or detect unintended use of reserved identifiers.

Related documentation

- [Basics on Generating Code Compliant with C99/C11/C++](#) (📖 [TargetLink Customization and Optimization Guide](#))
- [How to Specify Custom Identifier Restrictions](#) (📖 [TargetLink Customization and Optimization Guide](#))

Common subset of C90 and C++11/C++14/C++17

TargetLink 5.0 supports the common subset of C90 and C++11/14/17.

TargetLink 5.0 supports the generation of `extern C` encapsulation via the new `EmitEncapsulationByExternC` Code Generator option. This allows for integration of C production code into C++ projects.

Related documentation

- [Basics on Generating Code Compliant with C99/C11/C++](#) ( [TargetLink Customization and Optimization Guide](#))
- [EmitEncapsulationByExternC](#) ( [TargetLink Model Element Reference](#))

Void casts in function calls

Void casts in function calls - Generation of void casts for the suppression of return values

Via the new `InsertVoidCastForUnusedFcnCallReturnValue` Code Generator option, TargetLink 5.0 lets you specify whether a void cast is inserted in front of function calls if the return value of the function is not used.

Related documentation

- [InsertVoidCastForUnusedFcnCallReturnValue](#) ( [TargetLink Model Element Reference](#))

Generating Code for 64-Bit Linux Targets

TargetLink can now generate generic ANSI-C production code for 64-Bit Linux targets. In this context the base types of 32-Bit integer data types are defined as follows:

- `Int32`: `signed int` instead of `signed long int`.
- `UInt32`: `unsigned int` instead of `unsigned long int`.

The reason is that on 64-bit Linux targets the length of the long data type is 8 byte, on 32-bit Linux and Windows targets 4 bytes. Therefore, on 64-bit Linux `signed long int` and `unsigned long int` do not define 32-bit integer data types.

Improved Code Efficiency

Combining Code of If-Statements

TargetLink can now combine the code of several `if` statements that depend on the same or negated condition.

Related documentation

- [Combining Code of If Statements](#) ([📖 TargetLink Customization and Optimization Guide](#))
- [CombineControlFlowStatements](#) ([📖 TargetLink Model Element Reference](#))

Merging of nonconsecutive loops

TargetLink now tries to merge nonconsecutive loops. This even works when logging via log macros is enabled.

Related documentation

- [Merging Loops](#) ([📖 TargetLink Customization and Optimization Guide](#))
- [CombineControlFlowStatements](#) ([📖 TargetLink Model Element Reference](#))

Target Simulation (PIL)

Changes in the Target Simulation Modules

Support for new evaluation board

The Target Simulation Module of TargetLink 5.0 supports the Texas Instruments LAUNCHXL2570LC43 evaluation board.

Related documentation

- [Combinations of Evaluation Boards and Compilers](#) ([📖 Evaluation Board Reference](#))

Availability of support for Freescale MPC5604B

As of TargetLink 5.0, the Freescale MPC5604B evaluation board is supported via a Target Simulation Module (TSM) Extension instead of the TSM. TSM Extensions are free of charge if you have a valid Software Maintenance Service (SMS) contract.

Related documentation [Combinations of Evaluation Boards and Compilers](#) ([📖 Evaluation Board Reference](#))

New and discontinued compiler versions

The following table shows the compiler versions that are now supported by TargetLink 5.0. 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	—

Microcontroller Family	Compiler	New	No Changes	Discontinued
MPC57xxVLE	Diab	—	5.9	—
	GreenHill	2019	—	2018
MPC560xVLE	Diab	—	—	5.9
	GreenHill	—	—	2018
RH850	GreenHill	2019	—	2018
S12X	Cosmic	—	4.8	—
	Metrowerk	—	5.1	—
SH2	Renesas	—	9.3	—
SH2A-FPU	Renesas	—	9.4	—
TriCore17xx	TASKING	—	3.2	6.2
TriCore2xx	TASKING	6.3	—	6.2
	GCC	4.9	—	4.6
TMS570 (ARM)	CCS	7.0	—	—
V850	GreenHill	2019	—	2018
XC22xx	TASKING	—	3.0	—

For more information on the evaluation boards supported by TargetLink, refer to [Combinations of Evaluation Boards and Compilers \(Evaluation Board Reference\)](#).

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](#).

Usability

Where to go from here

Information in this section

Data Dictionary and Data Dictionary Manager	168
Property Manager	169

Data Dictionary and Data Dictionary Manager

Data Dictionary Manager Dialogs

TargetLink 5.0 provides the new **Advanced Edit String List** dialog in the Data Dictionary Manager.

In addition, the following Data Dictionary Manager dialogs are optimized in TargetLink 5.0:

- Edit String List
- Plain Variable
- Struct or Implicit Struct Variable

Related documentation

- Plain Variable ([📖 TargetLink Data Dictionary Manager Reference](#))
- Advanced Edit String List ([📖 TargetLink Data Dictionary Manager Reference](#))
- Edit String List ([📖 TargetLink Data Dictionary Manager Reference](#))
- Struct or Implicit Struct Variable ([📖 TargetLink Data Dictionary Manager Reference](#))

Import and Export Dialogs

TargetLink 5.0 provides improved DD import and export dialogs that stay open for repeated import and export. This lets you keep settings you already specified in the dialog.

Related documentation

- Import from AUTOSAR File ([📖 TargetLink Data Dictionary Manager Reference](#))
- Import from XML File ([📖 TargetLink Data Dictionary Manager Reference](#))
- Import from OIL File ([📖 TargetLink Data Dictionary Manager Reference](#))
- Export as AUTOSAR File ([📖 TargetLink Data Dictionary Manager Reference](#))
- Export as A2L File ([📖 TargetLink Data Dictionary Manager Reference](#))
- Export as XML File ([📖 TargetLink Data Dictionary Manager Reference](#))
- Export as OIL File ([📖 TargetLink Data Dictionary Manager Reference](#))

Usability improvements

TargetLink 5.0 provides several usability improvements in the Data Dictionary Manager, for example:

- Improved menu extensions
- A new details pane for objects and properties
- A symbol for custom properties
- An improved saving of DDs, write-protected DDs and DD include files

Related documentation

- Details ([📖 TargetLink Data Dictionary Manager Reference](#))
- Save As ([📖 TargetLink Data Dictionary Manager Reference](#))
- Save ([📖 TargetLink Data Dictionary Manager Reference](#))
- Save All ([📖 TargetLink Data Dictionary Manager Reference](#))
- Add Custom Property ([📖 TargetLink Data Dictionary Manager Reference](#))

- [How to Include Partial Data Dictionary Files](#) ([📖 TargetLink Data Dictionary Basic Concepts Guide](#))
- [Basics on Adding Custom Functionality to the Data Dictionary Manager](#) ([📖 TargetLink Data Dictionary Basic Concepts Guide](#))

Data Model Revision

TargetLink 5.0 provides data model revision numbers for included DD files and for XML import and export.

Related topics

- [Basics on Importing XML Files](#) ([📖 TargetLink Interoperation and Exchange Guide](#))
- [Basics of Exporting XML Files](#) ([📖 TargetLink Interoperation and Exchange Guide](#))

Adjust to Typedef for multiple DD objects at once

For a more efficient operation, the Adjust to Typedef functionality can now be used on multiple DD objects at once as follows:

- On DD VariableGroup objects to adjust all child objects.
- On multiple selected DD Variable objects and DD InterRunnableVariables objects to adjust all selected objects.

Related documentation

- None.

Property Manager

Validation Summary

TargetLink 5.0 provides a Validation Summary in the Property Manager. It displays model element data validation errors from all model element variables of the Property View. It lets you search, filter, and group errors.

Related documentation

- [Display of Validation Errors](#) ([📖 TargetLink Preparation and Simulation Guide](#))
- [Validation Summary](#) ([📖 TargetLink Tool and Utility Reference](#))
- [How to Search for Model Elements and Validation Errors](#) ([📖 TargetLink Preparation and Simulation Guide](#))
- [How to Filter for Property Values and Validation Errors](#) ([📖 TargetLink Preparation and Simulation Guide](#))
- [How to Customize Columns in the Property View and Validation Summary](#) ([📖 TargetLink Preparation and Simulation Guide](#))

Code Generator Options

New Code Generator Options

Overview of new Code Generator options

The following new Code Generator options are available with TargetLink 5.0.

- **CombineControlFlowStatements**
Combine two or more control flow statements with compatible controlling expression into one control flow statement. Refer to [Improved Code Efficiency](#) on page 165.
- **EmitEncapsulationByExternC**
Emit `extern C` encapsulation via `#ifdef __cplusplus`.
- **UtilizeBitwiseShiftOperations**
Use bitwise shift operations to implement efficient multiplications and divisions in a portable manner.
- **EmitAllTagNameConflictsAsError**
Emit an error if a tag name is simultaneously used as another identifier.
- **InsertVoidCastForUnusedFcnCallReturnValue**
Enable the insertion of void casts in front of function calls if the return value is not used.
- The following Code Generator options can be used to map Simulink® data types to TargetLink data types for local MATLAB® variables whose data type is not explicitly specified:
 - **AutoSelectBoolTypesForLocalMATLABVariables**
Map the Simulink® data type `logical` to the TargetLink data type `Bool`.
 - **AutoSelectIntegerTypesForLocalMATLABVariables**
Map the Simulink® data types `int*` and `uint*` to the equivalent TargetLink data type, e.g., `int16` <-> `Int16`, `uint8` <-> `UInt8`.
 - **AutoSelectFloatTypesForLocalMATLABVariables**
Map the Simulink® data types `single` and `double` to the equivalent TargetLink data types, e.g., `single` <-> `Float32`, `double` <-> `Float64`.

Refer to [Details on Determination and Mapping of Simulink® Data Types to TargetLink Data Types](#) ([TargetLink Code Generation Guide for MATLAB® Code in Simulink® Models](#)).

Related documentation

- [CombineControlFlowStatements](#) ([TargetLink Model Element Reference](#))
- [EmitEncapsulationByExternC](#) ([TargetLink Model Element Reference](#))
- [UtilizeBitwiseShiftOperations](#) ([TargetLink Model Element Reference](#))
- [EmitAllTagNameConflictsAsError](#) ([TargetLink Model Element Reference](#))
- [InsertVoidCastForUnusedFcnCallReturnValue](#) ([TargetLink Model Element Reference](#))

- [AutoSelectBoolTypesForLocalMATLABVariables](#) ([TargetLink Model Element Reference](#))
- [AutoSelectIntegerTypesForLocalMATLABVariables](#) ([TargetLink Model Element Reference](#))
- [AutoSelectFloatTypesForLocalMATLABVariables](#) ([TargetLink Model Element Reference](#))

For reference information on all Code Generator options, refer to [Alphabetical List of Code Generator Options](#) ([TargetLink Model Element Reference](#)).

Migration aspects of Code Generator options

For more information, refer to [Migration Aspects Regarding Code Generator Options](#) on page 184.

API Functions and Hook Scripts

Where to go from here

Information in this section

New API Functions	171
New Hook Scripts	172
Improved API Functions and Hook Scripts	172

New API Functions

List of new API functions

API Function	Purpose
tlFindHook (TargetLink API Reference)	Searches for TargetLink hook scripts whose names match the specified pattern.
tlRebuildFixedPointLibrary (TargetLink API Reference)	Rebuilds the Fixed-Point Library.
<code>dsdd('New'[,<template>])</code>	Creates a new DD workspace using a pre-defined or user template.

New Hook Scripts

New hook scripts

No new hook scripts have been introduced with TargetLink 5.0.

Improved API Functions and Hook Scripts

Improved API functions and hook scripts

With TargetLink 5.0, the following API functions and hook scripts have been improved.

tl_build_standalone and tl_tl2rti

With TargetLink 5.0, you can create a stand-alone TargetLink S-function without starting code generation via the new property `SkipCodeGeneration` of the `tl_build_standalone` and `tl_tl2rti` API function. By default, the new property is set to `off`.

To use this new property, you have to ensure that the required production code is available. The production code's description, i.e., the DD Subsystem object and the DD Application object and their child objects, must exist in the open Data Dictionary.

Related documentation

- `tl_build_standalone` ([📖 TargetLink API Reference](#))
- `tl_tl2rti` ([📖 TargetLink API Reference](#))

tlSimulinkBusObject

With TargetLink 5.0, the `tlSimulinkBusObject` API function can create a DD Typedef object from a Simulink.Bus object.

Related documentation

- `tlSimulinkBusObject` ([📖 TargetLink API Reference](#))
- [Basics on Preparing Simulink Systems for TargetLink Code Generation](#) ([📖 TargetLink Preparation and Simulation Guide](#))
- [Basics on Modeling Buses via Data Store Blocks](#) ([📖 TargetLink Preparation and Simulation Guide](#))

Other

Where to go from here

Information in this section

General Enhancements and Changes	173
TargetLink Demos	174

General Enhancements and Changes

Navigating TargetLink menus without using a mouse




In the Simulink Model Editor, you can now navigate in TargetLink menus via keyboard without using a mouse. First, press the **Alt** key to display the current menu with a unique letter underlined in each menu item. Then enter the related letter to select a specific menu item.

You can abort keyboard menu navigation by pressing **Alt** again.

Extracting model referencing subsystem via block dialog

With TargetLink 5.0, you can extract model referencing subsystems via the Extract subsystem button on the Incremental page of the Function Block block. This was already possible via the `t1ExtractSubsystem` API command.

Related documentation

- [Function Block](#) ( [TargetLink Model Element Reference](#))
- [Basics on Model Referencing](#) ( [TargetLink Customization and Optimization Guide](#))
- [How to Extract Subsystems to a Test Frame in a Separate Model](#) ( [TargetLink Customization and Optimization Guide](#))

Messages for conflicting specifications

The message E17299 is displayed if an identifier is not unique and does not meet the requirements for variable merging.

However, in TargetLink < 5.0 this message is displayed only once per code generation run, even if the model contains several identifiers where merging is not possible.

With TargetLink 5.0, this message is displayed multiple times if required. This reduces the number of code generation attempts, especially when merging variables.

TargetLink Demos

New demo models

The following new demo models are available for TargetLink 5.0:

- `AR_INCREMENTAL_CODEGEN` ([📖 TargetLink Demo Models](#))
- `VECU_EVENT_SIMULATION` ([📖 TargetLink Demo Models](#))

Migrating to TargetLink 5.0 and TargetLink Data Dictionary 5.0

Upgrade process Carefully read all of the following information and modify the tool chain accordingly.

Where to go from here	Information in this section
	General Migration Information 175
	Migrating from TargetLink 4.4 to 5.0 183
	Code Changes Between TargetLink 4.4 and TargetLink 5.0 193
	Discontinuations as of TargetLink 5.0 207

General Migration Information

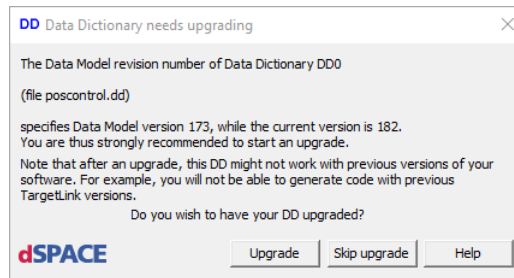
Upgrading Models, Libraries, and Data Dictionaries

Where to go from here	Information in this section
	Basics on Migrating Between TargetLink Versions 175
	How to Upgrade a Data Dictionary with Included DD Files 178
	How to Manually Upgrade Libraries and Models via the API 179
	Migrating Data Dictionaries to CodeDecorationSets 180

Basics on Migrating Between TargetLink Versions

Automatic upgrade from TargetLink 3.1 or later TargetLink 5.0 automatically upgrades models, TargetLink-compliant libraries, and Data Dictionaries if they were created with TargetLink 3.1 or later. You are

prompted for the automatic upgrade when the Data Dictionary is opened with TargetLink 5.0 for the first time. For example:



The automatic upgrade includes all the steps required by the intervening TargetLink versions. For example, an automatic upgrade from TargetLink 4.0 to TargetLink 5.0 comprises the steps 4.0 to 4.1 to 4.2 to 4.3 to 4.4.

Note

Check the TargetLink migration documentation of the different TargetLink versions to see 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, no upgrade can be performed.
- To upgrade DD files with included DD files, refer to [How to Upgrade a Data Dictionary with Included DD Files](#) on page 178.
- 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, no upgrade can be performed.

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 5.0:

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 later TargetLink versions because TargetLink automatically performs an upgrade. The library can still be used with TargetLink versions earlier than TargetLink 5.0 because the automatic upgrade does not save a library in the newer TargetLink version.

Only the current TargetLink version 5.0 is available Use TargetLink version 5.0 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 179. If you follow the instructions, the library is saved in TargetLink version 5.0 and, hence, cannot be used with TargetLink versions earlier than TargetLink 5.0.

Manual upgrade from TargetLink 2.x or 3.0.x

Models, libraries, and Data Dictionaries created with TargetLink versions 2.x or 3.0.x have 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

You cannot use models in the format of newer 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 ¹⁾
<code>dsdd_free;</code>	<code>dsdd_free;</code>
<code>dsdd('ReadFilterRuleSet', 'file', '<myFile>.xml');</code>	<code>dsdd('ReloadFilterRuleSets');</code>
<code>ds_error_register(dsdd('GetMessageList'));</code>	<code>ds_error_register(dsdd('GetMessageList'));</code>
<code>ds_msgdlg('update');</code>	<code>ds_msgdlg('update');</code>

¹⁾ All the files contained in the directory defined in Data Dictionary - Filter Rules in the Preferences Editor.

TargetLink informs you about errors in the TargetLink Message Browser. Each error contains the following information so that you can fix it in an XML-capable editor of your choice:

- File name
- 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](#) 179

References

[tlUpgrade](#) (📖 TargetLink API Reference)

How to Upgrade a Data Dictionary with Included DD Files

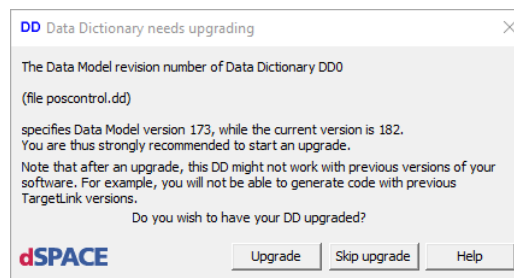
Precondition

In the main DD file to be loaded, the AutoLoad property of the DDIncludeFiles objects is set to on.

Method

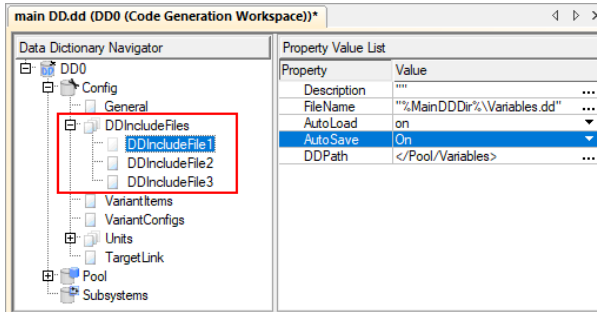
To upgrade a Data Dictionary with included DD files

- 1 Open the Data Dictionary Manager and the main DD via File - Open. The Data Dictionary needs upgrading dialog automatically opens if an earlier data model revision is involved.



- 2 Select Upgrade in the dialog. The main DD and all the included DD files are upgraded to the latest data model revision.

- Set the AutoSave property of all the /Config/DDIncludeFiles/<DDIncludeFile> objects to on to save all included DD files together with the main DD.



If you do not want to save all included DD files together with the main DD, specify their AutoSave property as required.

- Save the main DD file.

Result

You upgraded the main DD file and all its included DD files. TargetLink adjusted the revision number of every upgraded DD file to the latest data model revision.

Related topics

Basics

[Basics on Opening and Handling DD Files](#) (📖 TargetLink Data Dictionary Basic Concepts Guide)

HowTos

[How to Include Partial Data Dictionary Files](#) (📖 TargetLink Data Dictionary Basic Concepts Guide)

References

[Point of Inclusion](#) (📖 TargetLink Data Dictionary Manager Reference)

How to Manually Upgrade Libraries and Models via the API

Objective

To prepare a central upgrade of libraries and models in a tool chain scenario with several users, for example.

Preconditions

The model or library files are available on the MATLAB search path but they are not open.

The required and upgraded DD project file has been opened, for example, via `dsdd_manage_project('Open', '<name>.dd')`. DD project files can be upgraded via `dsdd('Upgrade'[, <DD_Identifier>])`.

Method

To manually upgrade libraries and models via the API

- 1 Type the following 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)

Migrating Data Dictionaries to CodeDecorationSets

Introduction of CodeDecorationSet and CodeDecoration objects

TargetLink 4.3 introduced DD CodeDecorationSet and CodeDecoration objects.

Additionally, several properties were removed from the Data Dictionary data model:

DD Object	Change	Replacement
FunctionClass	Removal of the DeclarationStatements and SectionName properties.	The DeclarationStatements and SectionName properties of the DD CodeDecoration.Settings object.
VariableClass		
VariableClassTemplate.Filter	Removal of the WidthSpec property.	The WidthSpec property of the DD CodeDecoration.Filter 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 the 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	DeclarationStatements or SectionName properties are set.	1. Creating a DD CodeDecorationSet object. 2. Creating a single DD CodeDecoration object for each DD CodeDecorationSet object. The settings of the CodeDecoration object and its child objects match the settings of the original objects. 3. Referencing the CodeDecorationSet object at the original object.
FunctionClass		
SubStructTemplate	Filter.VariableClass is set.	Transfer the values of the following properties from the variable class to the SubStructTemplate object's filter: <ul style="list-style-type: none"> ▪ DeclarationStatements ▪ SectionName ▪ TypePrefix
VariableClassTemplate	<ul style="list-style-type: none"> ▪ Filter.FilterCondition property is set to ALL_TRUE. ▪ Settings.VariableClass references a DD VariableClass object whose DeclarationStatements or SectionName properties are set. ▪ The Filter.WidthSpec property is set for this DD VariableClassTemplate object or for another VariableClassTemplate object whose Filter.VariableClassSpec property has the same value. 	1. Create a new DD VariableClass object in /Pool/VariableClasses/Templates. 2. Create a new DD CodeDecorationSet object in /Pool/CodeDecorations/Templates. 3. For each VariableClassTemplate object with the same value at the Filter.VariableClassSpec property, adding a CodeDecoration object to the CodeDecorationSet object. 4. Specifying the CodeDecoration object as required. 5. Referencing the CodeDecorationSet object at the VariableClass object created in step 1. 6. Referencing the VariableClass object created in step 1 via the VariableClassTemplate.Settings.VariableClass property.

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 the upgrade without replacement.

If you want to keep the property's value, set the DD VariableClassTemplate object's Filter.FilterCondition property 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

The automatic upgrade retains the functionality that was specified in the old Data Dictionary. You can clean it 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 the Data Dictionary. For example, if you used 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 DD `CodeDecoration` objects that were generated during the upgrade for each variable class called `<Name>_<width>` to 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 the original `<Name>_<width>` variable classes were referenced by variable class templates, the DD upgrade automatically creates 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

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 use them again 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 the SRC (highest scope) as required.
2. Delete the other variable classes of the SRC.
3. If you also used the 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 CodeDecoration objects can influence the generated production code mainly in the following respects:

- [Changed code comments](#) ( [TargetLink New Features and Migration Guide](#))
- [Sorting of variable definitions](#) ( [TargetLink New Features and Migration Guide](#))

Refer to [Code Changes Between TargetLink 4.3 and TargetLink 4.4](#) ( [TargetLink New Features and Migration Guide](#)).

Migrating from TargetLink 4.4 to 5.0

Where to go from here

Information in this section

Code Generator Options	184
AUTOSAR	187
Custom Code	188
API Functions and Hook Scripts	189
Messages	190
Other Migration Aspects	191

Code Generator Options

Migration Aspects Regarding Code Generator Options

Removed Code Generator options

The following Code Generator options were removed from TargetLink:

Removed Option	Additional Information
DisableFunctionsAsAnalysisBoundaries	-
ShiftMode	As a replacement, use the new UtilizeBitwiseShiftOperations (TargetLink Model Element Reference) option.
SideEffectFreeAnalysisThreshold	-
TreatAllForcedAtomicSubsystemsAsWeakAtomic	-
UseSfDataAsStateFlags	As a replacement, you can use active state data with child activity mode. Refer to Basics on Working with Active State Data (TargetLink Preparation and Simulation Guide).

Changed Code Generator options

Code Generator Option	Old Default	New Default
None	-	-

Recommended compatibility settings for UtilizeBitwiseShiftOperations

ShiftMode option	UtilizeBitwiseShiftOperations option ¹⁾	Additional Information
0 (Automatic)	1 (Automatic)	-
1 (Don't shift right)	4 (AvoidAnyNonPortableShiftsBasedOnSignedness)	Differences to ShiftMode: <ul style="list-style-type: none"> Right-shifts are allowed for operands with unsigned data types. Left-shifts are restricted to operands with unsigned data types.
17 (Don't shift right, keep explicit shift operations)		
2 (Don't shift left)	2 (AvoidPotentiallyUndefinedShiftsBasedOnSignedness)	Differences to ShiftMode: <ul style="list-style-type: none"> Left-shifts are allowed for operands with an unsigned data type. This is the default setting.
18 (Don't shift left, keep explicit shift operations)		
3 (Don't shift at all)	6 (Never)	If you want to avoid undefined or implementation-defined behavior, you can use option 4 as well.
19 (Don't shift at all, keep explicit shift operations)		

¹⁾ Explicitly modeled shift operations are taken into account. A message of the **Advice** type is displayed if an explicit shift operation does not comply with the Code Generator option.

With TargetLink 5.0, signed operands are no longer left-shifted by default. You can either allow TargetLink to utilize range information or set the **UtilizeBitwiseShiftOperations** option to **Automatic**. This extends or generally allows for the use of left shifts on signed operands. Refer to [Suppression of Shift Operations](#) ([TargetLink Preparation and Simulation Guide](#)).

Note

The change in default behavior and semantics between the Code Generator options **ShiftMode** and **UtilizeBitwiseShiftOperations** leads to changes in the production code. Refer to [Shift Operations](#) on page 194.

Other Compatibility Settings

CombineControlFlowStatements Set the new **CombineControlFlowStatements** Code Generator option to **Off** to ensure downward compatibility.

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 replacing the model-based option settings that was 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 were set to 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_{Old} to TargetLink_{New}) 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 _{Old}) Default = 9	Option Value (≤ TargetLink _{New})	
		Default = 9 (Scenario #1)	Default = 11 (Scenario #2)
Model-based	9 ¹⁾	9 ¹⁾	11 ²⁾
	11	11	11 ¹⁾
	13	13	13
DD-based	9	9	9 ³⁾
	11	11	11
	13	13	13

¹⁾ The option value is not stored with the model because it equals the default.
²⁾ Manual reset might be necessary.
³⁾ Manual adjustment might be necessary.

Option value = new default If the Code Generator options were not set to default values in the former TargetLink version (A) but are 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_A ⇒ TargetLink_B ⇒ TargetLink_C and upgrading TargetLink_A ⇒ TargetLink_C can cause different option values. Refer to the following table.

If the default values for TargetLink versions A, B, and C read 9, 11, and 13, and an option was set to 11 in version A, an upgrade to version C changes the option value as follows:

Upgrade Strategy	Option Value TargetLink _A Default = 9	Option Value TargetLink _B Default = 11	Option Value TargetLink _C Default = 13
A ⇒ B ⇒ C	11 (≠ default)	11 (= default) ¹⁾	13 (= default) ¹⁾
A ⇒ C	11 (≠ default)	—	11 (≠ default)

¹⁾ The option value is not stored with the model because it equals the default.

New Code Generator options For more information on new Code Generator options, refer to [New Code Generator Options](#) on page 170.

AUTOSAR

Where to go from here

Information in this section



AUTOSAR 2.x/3.x no Longer Supported	187
Changed Block Property for AUTOSAR Mode	188

AUTOSAR 2.x/3.x no Longer Supported

TargetLink no longer supports code generation for AUTOSAR 2.x/3.x. If you want to use TargetLink 5.0 to generate code for an old model, you must set the DD /Pool/Autosar/Config property from 2.x/3.x to >= 4.0. This results in code changes.

If this code is to be compliant with AUTOSAR 4.x, adjust your project accordingly. If you want to create application data types, TargetLink assists you with its ApplicationDataType Creation Wizard.

Related documentation

- [Supported AUTOSAR Releases](#) ( [TargetLink Interoperation and Exchange Guide](#))
- [Code Changes Between TargetLink 4.4 and TargetLink 5.0](#) on page 193
- [How to Create Application Data Types for Existing Implementation Data Types](#) ( [TargetLink AUTOSAR Modeling Guide](#))

Changed Block Property for AUTOSAR Mode

With the support of Adaptive AUTOSAR, the `autosar.useautosarcommunication` block property was replaced by the `autosarmode` block property. This is shown in the following table:

TargetLink Version	Block Property	Data type
≤ 4.4	<code>autosar.useautosarcommunication</code>	Boolean (TargetLink API Reference)
5.0	<code>autosarmode</code>	AutosarModeEnum (TargetLink API Reference)

Adapt your user scripts and tool chain accordingly.

Custom Code

Dereferencing of Pointers

New behavior

With prior TargetLink versions it could be required to manually dereference pointers in custom code file sections. This no longer is the case: TargetLink now automatically determines whether a pointer in a custom code file section must be dereferenced by analyzing the context in which the Custom Code block resides.

Required migration of manually dereferenced pointers

If you used specialized custom code files whose custom code sections contain manually dereferenced pointers, you have to adjust these files to TargetLink's new behavior.

The following table show cases that have to be adapted to prevent false code (double dereferencing):

Expression in Custom Code File Section	Pointer	TargetLink ≤ 4.4	TargetLink 5.0
<code>y = u + *p;</code>	<code>Sa8_CC_param</code>	<code>Sa8_CC_out = Sa8_CC_in + *Sa8_CC_param;</code>	<code>Sa8_CC_out = Sa8_CC_in +>(*Sa8_CC_param);</code> ¹⁾
<code>x = p->x0 * u;</code>	<code>pPar</code>	<code>Sb3_CC_x = pPar->x0</code>	<code>Sb3_CC_x = (*pPar)->x0</code> ¹⁾

¹⁾ False code that contains double dereferencing.

Recommended practice

We strongly recommend to let TargetLink decide whether dereferencing is required. This is shown in the following table:

Expression in Custom Code File Section	Pointer	TargetLink ≤ 4.4	TargetLink 5.0
<code>y = u + p;</code>	Sa8_CC_param	<code>Sa8_CC_out = Sa8_CC_in + Sa8_CC_param;</code> ¹⁾	<code>Sa8_CC_out = Sa8_CC_in + (*Sa8_CC_param);</code> ²⁾
<code>y = u + p;</code>	-	<code>Sa8_CC_out = Sa8_CC_in + Sa8_CC_param;</code>	<code>Sa8_CC_out = Sa8_CC_in + Sa8_CC_param;</code>
<code>x = p.x0 * u;</code>	pPar	<code>Sb3_CC_x = pPar.x0</code> ¹⁾	<code>Sb3_CC_x = (*pPar).x0</code> ²⁾

¹⁾ False code with missing dereferencing.

²⁾ Correct code with dereferencing.

API Functions and Hook Scripts

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

Changes in API Functions	189
Changes in Hook Scripts	190

Changes in API Functions

tlOperationMode

With TargetLink 5.0, the `tlOperationMode` API uses the following values for the `OperationMode` parameter:

- `ModelingOnly`
- `FullFeatured`

For the easy adjustment of scripts, TargetLink 5.0 introduces the new command `IsFullFeatured`. It is recommended to adjust the scripts as shown in the following example:

TargetLink < 5.0	TargetLink 5.0
<code>if (strcmp(tlOperationMode('get'), 'full-featured'), ...</code>	<code>if tlOperationMode('IsFullFeatured') ...</code>

Related documentation

- `tlOperationMode` ([📖 TargetLink API Reference](#))

Changes in Hook Scripts

Changes in Hook Scripts

No hook scripts have to be migrated to TargetLink 5.0.

Messages

Message changes

Changed message for keyword use

With TargetLink 5.0, a different error message is displayed if you use a C90 keyword as identifier:

TargetLink < 5.0	TargetLink 5.0
E17056	E17553

Related documentation

- [Basics on Generating Code Compliant with C99/C11/C++](#) ( [TargetLink Customization and Optimization Guide](#))

VariableClass properties


In TargetLink < 5.0, invalid properties of DD VariableClass objects lead to a warning and the invalid property is ignored during code generation.

With TargetLink 5.0, invalid properties of DD VariableClass objects lead to an error.

TargetLink < 5.0	TargetLink 5.0
W17293	E17555

For components of explicit structs, the properties Const, Volatile, and TypePrefix are determined by the specification of the DD TypedefComponent objects of the struct Typedef, not the respective properties of the DD VariableClass object.

Related documentation

- [Basics on Arranging Data in Structured Variables](#) ( [TargetLink Customization and Optimization Guide](#))

Other Migration Aspects

Various Migration Aspects

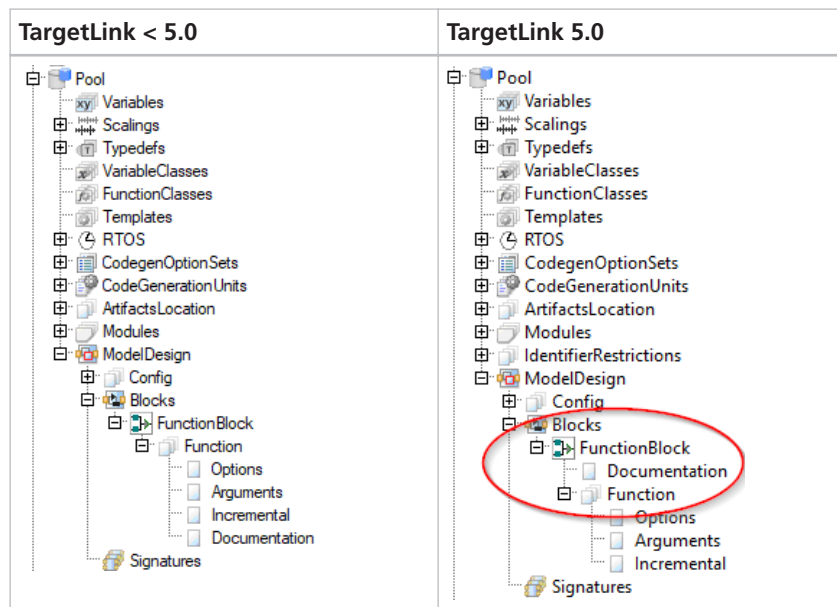
DD Documentation objects

DD Documentation objects changed:

In TargetLink < 5.0, the DD Documentation object is a child object of the DD Function object, that is a child object of the DD Block object. DD Documentation objects were available for DD Block objects whose BlockType property is set to TL_Function.

TargetLink 5.0 introduces new DD Documentation objects that are of object kind BlockDocumentation. The new objects are child objects of DD Block objects.

This leads to differences in the API functions. Refer to [Obsolete API Functions](#) on page 209.



The new DD Documentation objects of object kind BlockDocumentation are available for all DD Block objects, regardless of the value of the BlockType property. They will be evaluated for DD Block objects whose BlockType property is set to TL_Function or DataStoreMemory.

The new objects have the new CodeCommentPlacement property.

Related documentation

- [Basics on the Data Model as Shown in the DD Object Tree](#) ([TargetLink Data Dictionary Basic Concepts Guide](#))
- [Obsolete API Functions](#) on page 209

Stateflow paths

With TargetLink 5.0, the syntax for paths of Stateflow elements changes for the following API commands:

- `dsdd_get_block_path`
- `dsdd('GetPathToSLObject', ...)`

Elements are now separated by slashes ('/') as shown in the following example:

```
TL < 5.0
'sf_demo/sf_control/Subsystem/sf_control/Chart.do_blink.BLINKER_ON'

TL 5.0
'sf_demo/sf_control/Subsystem/sf_control/Chart/do_blink/BLINKER_ON'
```

Related documentation

- [dsdd_get_block_path](#) ([📖 TargetLink API Reference](#))
- Documentation on `dsdd('GetPathToSLObject', ...)` available only in dSPACE Help.

Inhomogeneous scalings and constrained ranges

Variables that describe the matrices of the Discrete State-Space block can have inhomogeneous scalings and inhomogeneous constrained ranges. With TargetLink 5.0, this is only possible if all of the following conditions hold:

- The variables do not reference DD Variable objects.
- The variables do not reference DD ReplaceableDataItem objects.
- The block property `Keep matrix structure` is not set.

In this case, TargetLink generates a structured variable for the Discrete State-Space block whose scalar components contain the values of the matrices.

Related documentation

- [Code Generation That Keeps Coefficient Vectors/Matrices in Struct Variables](#) ([📖 TargetLink Preparation and Simulation Guide](#))

ArbitraryLSB and autoscaling

Controlling the autoscaling mode for each block is now realized with the two new autoscaling modes `Calculate power-of-two scaling` and `Calculate arbitrary scaling`. These modes can be set via the `Logging & Autoscaling` page of the respective block or via the `autoscaling.mode` property of the API.

The TargetLink block property `variable.arb` is no longer available. Instead, TargetLink 5.0 lets you set the LSB to an arbitrary value without adjusting the `ArbitraryLSB` property beforehand. In the block dialog, the `2arb` button still conveniently lets you enter power-of-two scaling factors and converts scaling factors between decimal and power-of-two scaling.

Models are migrated automatically.

If you want to check the scaling of a block via scripts, you are recommended to test if the value of the `.lsb` property can be represented as a power of two as shown in the following example:

```
log2Value = log2(lsb);
isPowerOfTwo = round(log2Value) == log2Value;
```


Related documentation

- [Basics on Autoscaling of Output Variables](#) ([📖 TargetLink Preparation and Simulation Guide](#))

Fixed-Point Library

With TargetLink 5.0, the Fixed-Point Library is no longer delivered as a pre-compiled library. Instead, it will be compiled automatically during the build process in the following cases:

- The first time a build process for a model is started if the generated code depends on the Fixed-Point Library.
- Each time the source files for the library have changed.

In these cases, the compilation of the Fixed-Point Library leads to increased build times.

Additionally, you can manually start the compilation via the new API function `tlRebuildFixedPointLibrary`.

Related documentation

- [Basics on Specifying the Location of the Sources and Binaries of the TargetLink Fixed-Point Library](#) ([📖 TargetLink Customization and Optimization Guide](#))
- `tlRebuildFixedPointLibrary` ([📖 TargetLink API Reference](#))

Code Changes Between TargetLink 4.4 and TargetLink 5.0

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

Shift Operations	194
Extern C Encapsulation	196
Access Functions	196
AUTOSAR	197
Global Bitfields	198
MATLAB Code	199
Name Template of DD VariableTemplate Objects	199
Look-Up Tables	200
Variable Vector Width	201
Stateflow	201
Other	204

Shift Operations

Suppression of implicit shift operations

For shift operations that are not specified explicitly, the behavior of TargetLink changed. The associated `ShiftMode` Code Generator option was replaced by the `UtilizeBitwiseShiftOperations` Code Generator option that has different semantics and a different default behavior. Signed operands are no longer left-shifted by default. This leads to changes to the production code as multiplications are used more often. These changes are described below.

Reason Compliance with C99.

Migration issue For recommended compatibility settings for the new Code Generator option, refer to [Migration Aspects Regarding Code Generator Options](#) on page 184.

Multiplications with power-of-two values

Multiplications with power-of-two values are often used for rescaling operations. For signed operands, multiplications with power-of-two values are now performed as multiplications instead of left shifts by default. This is shown in the following example:

Operation:

```
Int16 I16Out; Lsb = 2^-1
Int16 I16In; Lsb = 2^0
I16Out = I16In;
```

Result:

TargetLink ≤ 4.4	TargetLink 5.0
<code>I16Out = I16In << 1;</code>	<code>I16Out = I16In * 2;</code>

Note

Most compilers implement multiplications with power-of-two values as efficient as shift operations.

Saturation

The saturation of shift operations is implemented with macros. The saturation of multiplications with a numerical value is implemented as control flows. With the new default behavior, the replacement of shift operations with multiplications therefore leads to the replacement of macro calls with control flow expressions. This is shown in the following example:

TargetLink ≤ 4.4	TargetLink 5.0
<pre>HOLD_VALUE = C_I16SHLI16C6_SATb(Sb15_Switch3, 5, 1023, -1024);</pre>	<pre>if (Sb15_Switch3 > 1023) { HOLD_VALUE = 32767; } else { if (Sb15_Switch3 < -1024) { HOLD_VALUE = -32768; } else { HOLD_VALUE = Sb15_Switch3 * 32; } }</pre>

Casts

Casts for operations with left shifts are different from casts for multiplications. This leads to changes in the production code as left shifts of signed operands are now avoided by default:

TargetLink ≤ 4.4	TargetLink 5.0
<pre>I16Out = (Int16) (I16Var1 << 1);</pre>	<pre>I16Out = I16Var1 * 2;</pre>

The missing final cast does not interfere with the MISRA C-compliance of the production code.

Signed operands and unsigned result

Expressions with a signed operand and unsigned results, e.g., code patterns for the Abs block, can change because of the different default behavior:

TargetLink 5.0
<pre>if (In >= 0) { OutAbs = (UInt8) (((UInt16) InI16) * 2); } else { OutAbs = (UInt8) (((UInt16) (-InI16)) << 1);}</pre>

In case of the `then` branch, the multiplication pattern casts the signed operand to unsigned. By default, TargetLink does not implement a shift in this expression because the signed operand is detected.

In case of the `else` branch, shifting is possible because the result of the unary minus operation already has an unsigned type.

Additions and subtractions with the same shift factor

Under certain conditions, additions and subtractions with the same shift factor were implemented by first performing the addition/subtraction and then shifting. This is different for multiplications that by default now partly replace shift operations:

Operation:

```
out = in1 << 1 + in2 << 1
```

Result:

TargetLink ≤ 4.4	TargetLink 5.0
<pre>out = (in1 + in2) << 1</pre>	<pre>out = in1 * 2 + in2 * 2</pre>

Constant folding

The change in default behavior can lead to more constant foldings because multiplications can be folded by TargetLink. Under certain conditions, overflows will be computed directly:

Operation	Result
<code>I16Var = (Int16)(32767 << 1);</code>	<code>I16Var = -2;</code>
<code>I16Var = (Int16)(-32768 << 1);</code>	<code>I16Var = 0;</code>

Operand eliminations

Shift operations that eliminate operands can now be replaced by multiplications that do not eliminate operands as shown in the following example:

Operation:

```
U16Var = I16Var * 65536
```

Result:

TargetLink ≤ 4.4	TargetLink 5.0
<code>U16Var = 0;</code>	<code>U16Var = I16Var * 65536</code>

Extern C Encapsulation

Production code now contains encapsulation of extern C:

TargetLink ≤ 4.4	TargetLink 5.0
-	<pre>#ifndef __cplusplus extern "C" { #endif ... // TargetLink-generated-C-Code #ifdef __cplusplus } #endif</pre>

Reason Lets you use TargetLink-generated production code in C++ contexts.

Migration issue You can control this behavior via the `EmitEncapsulationByExternC` Code Generator option.

Access Functions

Access functions containing only read accesses

The code pattern changed for access functions that contain only read accesses:

- Scalar variables: TargetLink now generates auxiliary variables to buffer the value more often if the number of accesses is greater than two

- Vector and matrix variables: TargetLink now generates an auxiliary variable to buffer the value only if the whole variable is accessed at least once
 - If you force TargetLink to use auxiliary variables by setting the `CreateLocalValueCopy` property to `always`, the following pattern is used: Preceding each read access of an element of the variable, TargetLink now uses an access function call to write the respective element.

Reason Code efficiency, Increased consistency

Migration issue None

Access functions specified as ADDRESS or AsAddressParameter

Access functions whose `PassVariable` property is set to `AsAddressParameter` now use the TargetLink base type.

TargetLink ≤ 4.4	TargetLink 5.0
<code>MyInt16 * GetMyAfArray3Var3(MyArrayV_3 * varptr)</code>	<code>Int16 * GetMyAfArray3Var3(Int16 * varptr)</code>

In calls of RTE API functions of classic AUTOSAR that are generated via access function specifications, the application data type is used:

TargetLink ≤ 4.4	TargetLink 5.0
<code>sint32 * Rte_Pim_SHZ_A4_Matrix(void)</code>	<code>MyInt32 * Rte_Pim_SHZ_A4_Matrix(void)</code>

Reason Increased consistency

Migration issue None

AUTOSAR

AUTOSAR 2.x/3.x no longer supported

TargetLink no longer supports code generation for AUTOSAR 2.x/3.x. After you changed the value of the `DD /Pool1/Autosar/Config` property from `2.x/3.x` to `>= 4.0`, the following code changes occur:

Evaluation of \$(Prm)

TargetLink ≤ 4.4	TargetLink 5.0
<code>Rte_Calprm...()</code>	<code>Rte_Prm...()</code>

Changed default for function class of runnables For DD Runnable objects that do not reference a DD FunctionClass object, the default function class has changed. This can result in a different order of declarations or definitions.

Additionally, there are new instructions for memory mapping:

TargetLink ≤ 4.4	TargetLink 5.0
-	<pre>#define Swc_Start_Sec_Code #include Swc_MemMap.h ... #define Swc_Stop_Sec_Code #include Swc_MemMap.h</pre>

And changes in compiler abstraction:

TargetLink ≤ 4.4	TargetLink 5.0
FUNC(void, RTE_APPL_CODE) Run(void)	FUNC(void, Swc_CODE) Run(void)

Removed Ptr2ArrayBasetypePassing macro

TargetLink ≤ 4.4	TargetLink 5.0
<pre>#ifndef RTE_PTR2ARRAYBASETYPE_PASSING #define RTE_PTR2ARRAYBASETYPE_PASSING 1 #endif</pre>	-

Reason Removal of code generation for AUTOSAR 2.x/3.x

Migration issue Refer to [AUTOSAR 2.x/3.x no Longer Supported](#) on page 187.

Global Bitfields

Boolean instead of UInt8

Boolean variables (UseGlobalBitfields = on) or Stateflow states and flags (StateflowUseBitfields = on) that are collected in global bitfields are now optimized differently:

TargetLink ≤ 4.4	TargetLink 5.0
Function local variables are removed from the global bitfield as UInt8	Function local variables are removed from the global bitfield as Boolean

This can result in a different number of casts, depending on the context.

Reason Bug fix

Migration issue None

Inheritance of data type

The data type of block variables might change if all of the following conditions are fulfilled:

- The UseGlobalBitfields Code Generator option is set to on
- The block's Inherit properties checkbox is selected
- The block has a variable class other than default
- The block output is scalar

- The block is driven by a block whose variable class is set to **default** and whose data type is **Bool**

TargetLink ≤ 4.4	TargetLink 5.0
The block variable inherits the data type Bitfield	The block variable inherits the data type Bool

This can also affect auxiliary variables that are generated in this context as well as their casts.

Reason Increased consistency

Migration issue None

MATLAB Code

Mapping of Simulink data types

When generating code for unspecified local MATLAB variables, TargetLink now determines the data type automatically. This can lead to type changes.

Migration issue You can control data type mapping via the following Code generator options:

- [AutoSelectIntegerTypesForLocalMATLABVariables](#) (📖 TargetLink Model Element Reference)
- [AutoSelectBoolTypesForLocalMATLABVariables](#) (📖 TargetLink Model Element Reference)
- [AutoSelectFloatTypesForLocalMATLABVariables](#) (📖 TargetLink Model Element Reference)

Name Template of DD VariableTemplate Objects

The default value of the NameTemplate property of DD VariableTemplate objects changed for global interface variables that are specified as follows:

- DD VariableKind property set to **GlobalInterfaceVar** and DD VariableClassSpec property set to **SLGlobal**.
- DD VariableKind property set to **GlobalInterfaceVar** and DD VariableClassSpec property set to **SLGlobalInit**.

TargetLink ≤ 4.4	TargetLink 5.0
IF_\$\$\$_L	IF_\$\$\$_B

This also affects implicitly generated struct components of global bitfield structures and reuse structures.

Additionally, in this context, **\$L** is now always evaluated as **\$B**.

Reason Increased consistency

Migration issue None

Look-Up Tables

Look-up table functions and table maps

Table functions and table maps now always use the TargetLink base type.

Reason Increased consistency

Migration issue None

Overflows of Direct Look-Up Table blocks

If the Action for out-of-range input property of Direct Look-Up Table (n-D) blocks is set to **Warning** or **Error**, TargetLink instruments the generated code to report overflows in SIL and PIL simulations. This could lead to compiler errors in the following circumstances:

- When compiling the production code outside of TargetLink
- The Global logging option was set to **Do not log anything** and the Clean code checkbox was cleared

With TargetLink 5.0, this has been fixed, which leads to code changes.

The generated macros were renamed:

TargetLink ≤ 4.4	TargetLink 5.0
<pre>ERR_LUT_INDEX_OVERFLOW ERR_LUT_INDEX_UNDERFLOW WARN_LUT_INDEX_OVERFLOW WARN_LUT_INDEX_UNDERFLOW</pre>	<pre>ERR_LUT_IDX_BEYOND_UPPER_LIMIT ERR_LUT_IDX_BEYOND_LOWER_LIMIT WARN_LUT_IDX_BEYOND_UPPER_LIMIT WARN_LUT_IDX_BEYOND_LOWER_LIMIT</pre>

Additionally, this results in changes in `tl_defines.h`:

Change	Example
<p><code>tl_defines.h</code> can now contain empty macro definitions that are encapsulated by preprocessor directives.</p>	<pre>#ifndef TL_FRAME #ifndef WARN_LUT_IDX_BEYOND_LOWER_LIMIT #define WARN_LUT_IDX_BEYOND_LOWER_LIMIT(param_0) #endif /* WARN_LUT_IDX_BEYOND_LOWER_LIMIT */ #endif /* TL_FRAME */ #ifndef TL_FRAME #ifndef WARN_LUT_IDX_BEYOND_UPPER_LIMIT #define WARN_LUT_IDX_BEYOND_UPPER_LIMIT(param_0) #endif /* WARN_LUT_IDX_BEYOND_UPPER_LIMIT */ #endif /* TL_FRAME */</pre>

Change	Example
t1_defines.h can now contain an encapsulated include of the simulation frame header file.	<pre>#ifdef TL_FRAME #include "TL_Controller_frm.h" #endif</pre>

Reason Bug fix

Migration issue None

Variable Vector Width

Names and order of auxiliary variables

The name and order of auxiliary variables can change in the context of variable vector width (VWV). This can be the case if these auxiliary variables are used in restart functions that are used to initialize VWV variables.

TargetLink ≤ 4.4	TargetLink 5.0
<pre>q_BasicColorVWVUDState[Aux_S32] = Temp_[Aux_S32]; r_BasicColorVWVUDState[Aux_S32] = Temp__a[Aux_S32]; m_LightColorVWVUDState[Aux_S32] = Temp__b[Aux_S32];</pre>	<pre>q_BasicColorVWVUDState[Aux_S32] = Temp__b[Aux_S32]; r_BasicColorVWVUDState[Aux_S32] = Temp__c[Aux_S32]; m_LightColorVWVUDState[Aux_S32] = Temp__h[Aux_S32];</pre>

Reason Bug fix

Migration issue None

Stateflow

Removal of UseSfDataAsStateFlags Code Generator option


With prior TargetLink versions you were able to use user-defined Stateflow data variables as state activity variables, if the UseSfDataAsStateFlags Code Generator option was enabled. These user defined state activity variables were called *user state flags*. With the removal of the UseSfDataAsStateFlags Code Generator option, user state flags are no longer supported. Instead, TargetLink generates an implicit state activity variable, as shown in the following example:

TargetLink ≤ 4.4	TargetLink 5.0
<pre>/* Begin execution of state Subsystem/Chart.StateA */ switch (MyUSF_StateA) { case 1: { ... } case 2: { ... } default: { break; } }</pre>	<pre>/* Begin execution of state Subsystem/Chart.StateA */ switch (SIBFS_Chart_a.Ca1_Chart_ns) { case StateA1_id: { ... } case StateA2_id: { ... } default: { break; } }</pre>

If your model contains a Stateflow data variable that was used as a user state flag, TargetLink displays message W21039.

Reason Resolves differences in MIL/SIL/PIL simulation modes

Replaced by support of Stateflow active state data with the **Child activity** activity type.

Migration issue As a replacement, you can use active state data with child activity mode. Refer to [Basics on Working with Active State Data](#) ( [TargetLink Preparation and Simulation Guide](#)).

State activity output and signal merging

In Stateflow you can output an active state data to Simulink via a port. If this port is connected to a Merge block, this might lead to false code. With TargetLink 5.0, in this context, the variable generated for the active state data is no longer used as a state activity variable:

TargetLink ≤ 4.4	TargetLink 5.0
<pre> ... switch (Ca1_AMode) { case AModeType_A1: { /* Begin execution of state TL_Root/Chart.A.A1 */ if (...) { /* State transition from TL_Root/Chart.A.A1 to TL_Root/Chart.A.A2 */ Ca1_AMode = AModeType_A2; Sa1_Merge_A = Ca1_AMode; } else { ... } /* End execution of state TL_Root/Chart.A.A1 */ break; } case AModeType_A2: { /* Begin execution of state TL_Root/Chart.A.A2 */ if (...) { /* State transition from TL_Root/Chart.A.A2 to TL_Root/Chart.A.A1 */ Ca1_AMode = AModeType_A1; Sa1_Merge_A = Ca1_AMode; } else { ... } /* End execution of state TL_Root/Chart.A.A2 */ break; } default: { break; } } ... </pre>	<pre> ... switch (SIBFS_Chart_a.Ca2_A_ns) { case Ca3_A1_id: { /* Begin execution of state TL_Root/Chart.A.A1 */ if (...) { /* State transition from TL_Root/Chart.A.A1 to TL_Root/Chart.A.A2 */ SIBFS_Chart_a.Ca2_A_ns = Ca4_A2_id; Ca1_AMode = AModeType_A2; Sa1_Merge_A = Ca1_AMode; } else { ... } /* End execution of state TL_Root/Chart.A.A1 */ break; } case Ca4_A2_id: { /* Begin execution of state TL_Root/Chart.A.A2 */ if (...) { /* State transition from TL_Root/Chart.A.A2 to TL_Root/Chart.A.A1 */ SIBFS_Chart_a.Ca2_A_ns = Ca3_A1_id; Ca1_AMode = AModeType_A1; Sa1_Merge_A = Ca1_AMode; } else { ... } /* End execution of state TL_Root/Chart.A.A2 */ break; } default: { break; } } ... </pre>

Reason Bug fix

Migration issue None

Removal of superfluous consecutive assignments

Unoptimized Stateflow code could contain superfluous consecutive assignments to variables that are internally used by TargetLink. With TargetLink 5.0, only the last assignment is generated.

TargetLink ≤ 4.4	TargetLink 5.0
<pre> SIBFS_Chart_a.Aux_Ca1_sflag0 = 0; SIBFS_Chart_a.Aux_Ca1_sflag0 = 1; </pre>	<pre> SIBFS_Chart_a.Aux_Ca1_sflag0 = 1; </pre>

Reason Suppress superfluous code

Migration issue None

Other

Changed default value of UseName property

The default value of the DD UseName property of DD VariableClass objects is now **off**. This affects only the following:

- DD VariableClass objects that you create in existing Data Dictionaries
- All the DD VariableClass objects in the preconfigured DD templates

TargetLink ≤ 4.4	TargetLink 5.0
<pre> /*****\ CAL: global calibratable parameters (ROM) Width: 16 *****/ CAL Int16 Sa1_Constant = 1; </pre>	<pre> /*****\ CAL: global calibratable parameters (ROM) Width: 16 *****/ const volatile Int16 Sa1_Constant = 1; </pre>

Additionally, includes might be placed in C files that were previously placed in H files.

Reason The macros generated for VariableClass objects are placed in the CGU-specific tl_defines header. Because variable classes are often used across several CGUs, this leads to multiple macro definitions. This includes the risk that the same variable class name is associated with different properties, leading to different macro definitions which is a C language constraint violation.

Migration issue None

Initialization of subsystem in ports and out ports

If a InPort or OutPort block of a subsystem references a DD Variable object that provides an initial value and whose Scope property is set to **value_param**, **ref_param**, or **fcn_return**, the initialization of the actual parameter in the production code changed:

TargetLink ≤ 4.4	TargetLink 5.0
<p>The actual parameter was initialized only in the following cases:</p> <ul style="list-style-type: none"> ▪ The code was generated for the OutPort block of conditionally executed subsystem. ▪ The code was generated for an InPort block that had one of the following Simulink checkboxes selected: <ul style="list-style-type: none"> ▪ Latch input by delaying the outside signal ▪ Latch input for feedback signals of function-call subsystem outputs 	<p>The actual parameter is now always initialized with the initialization value that is specified in the Data Dictionary.</p>

Reason Bug fix

Migration issue To revert to the old behavior, remove the initial value that is specified in the Data Dictionary.

Scope reduction of matrix variables

During optimization, the scope and storage duration of matrix variables is reduced earlier. This can result in a different order of optimizations.

Additionally, the following changes are possible:

- Fewer casts
- More inlining
- Auxiliary variables lose their suffix (`Aux_` instead of `Aux_b`)

Reason Code efficiency

Migration issue None

Function inlining

During code generation, TargetLink now uses a different strategy to inline functions. This is done to generate code that maximizes the number of functions of the same size:

- Earlier inlining of Stateflow functions.
- Inlining of functions that contribute to indirect recursion (Stateflow and MATLAB code)

This can lead to the following code changes:

- Additional casts in compute-through-overflow-calculations.
- Functions that were inlined by previous TargetLink versions are no longer inlined.
- Functions that were not inlined by previous TargetLink versions now are inlined.
- Variables are eliminated or reduced to function-local-scope. Assignments are moved into conditionally executed control flows.
- Variables that were reduced to function-local-scope in previous TargetLink versions are no longer reduced in scope.

Reason Increased consistency, Code efficiency

Migration issue None

Suffixes of auxiliary variables

In the context of inlined functions, the suffixes of auxiliary variables might change.

Reason Increased consistency

Migration issue None

Active variant config in file header comment

In previous TargetLink versions, the file header comment did not contain the active variant config if the `cconfig.xml` file contained the following lines:

```
<TL:header-comment show="true" show-in-non-root-files="true">
  ...
  <TL:active-variant show="true"/>
  ...
  <TL:option-list-comment show="false"/>
  ...
</TL:header-comment>
```

With TargetLink 5.0 this works as expected:

TargetLink ≤ 4.4

```

/*****\
***
*** Simulink model      : onLine_parameter_modification
*** TargetLink subsystem : onLine_parameter_modification/LissajousFigure
*** Codefile           : LissajousFigure.c
***
*** Generated by TargetLink, the dSPACE production quality code generator
***
*** SUBSYS             CORRESPONDING SIMULINK SUBSYSTEM
*** Sa1                LissajousFigure
***
*** SUBSYS             CORRESPONDING MODEL BLOCK (REFERENCED MODEL)
***
*** SF-NODE            CORRESPONDING STATEFLOW NODE          DESCRIPTION
***
\*****/

```

TargetLink 5.0

```

/*****\
***
*** Simulink model      : onLine_parameter_modification
*** TargetLink subsystem : onLine_parameter_modification/LissajousFigure
*** Codefile           : LissajousFigure.c
***
*** Generated by TargetLink, the dSPACE production quality code generator
***
*** Active Variant Config : VC1
***
*** SUBSYS             CORRESPONDING SIMULINK SUBSYSTEM
*** Sa1                LissajousFigure
***
*** SUBSYS             CORRESPONDING MODEL BLOCK (REFERENCED MODEL)
***
*** SF-NODE            CORRESPONDING STATEFLOW NODE          DESCRIPTION
***
\*****/

```

Reason Matches user expectations

Migration issue None

Different order of function declarations in <cg_u>_fri.h

In the simulation code of previous TargetLink versions, the <cg_u>_fri.h file could contain double declarations of the following kind of functions:

- AUTOSAR restart functions
- AUTOSAR terminate functions
- Access functions

With TargetLink 5.0 this was fixed. This can result in a different order of function declarations in the <cg>_fri.h file (simulation code):

TargetLink ≤ 4.4	TargetLink 5.0
<pre>extern void TerminateRun(void); extern void RESTART_SWC1_Run1(void); extern void Subsystem(void); ... extern void Task_RestartRun(void); extern void RestartRun(void); extern void TerminateRun(void); extern void GetSa2_In1(sint16 * valueptr); extern void SetSa2_Out1(const sint16 * valueptr); #endif /* SUBSYSTEM_FRI_H */</pre>	<pre>extern void GetSa2_In1(sint16 * valueptr); extern void RestartRun(void); extern void SetSa2_Out1(const sint16 * valueptr); extern void TerminateRun(void); extern void RESTART_SWC1_Run1(void); extern void Subsystem(void); ... extern void Task_RestartRun(void); #endif /* SUBSYSTEM_FRI_H */</pre>

Additionally, for the same kind of functions, function class comments or block declaration statements could be missing in production code. This also has been fixed:

TargetLink ≤ 4.4
<pre>FUNC(void, FuelsysController_CODE) FuelsysControllerInit(void);</pre>
TargetLink 5.0
<pre>/* start of memory section 'CODE' */ #define FuelsysController_START_SEC_CODE #include "FuelsysController_MemMap.h" /*****\ ARRunnable4: Default function class for AUTOSAR runnables *****/ FUNC(void, FuelsysController_CODE) FuelsysControllerInit(void); /* end of memory section 'CODE' */ #define FuelsysController_STOP_SEC_CODE #include "FuelsysController_MemMap.h"</pre>

Reason Bug fix
Migration issue None

Discontinuations as of TargetLink 5.0

Where to go from here

Information in this section

Discontinued TargetLink Features	208
Obsolete API Functions	209
Obsolete Limitations	210

Discontinued TargetLink Features

Code generation for special OSEK versions

The code generation for special OSEK versions, such as OsCan, is no longer supported with TargetLink 5.0 :

- The RTOS selection on the RTOS page of the [TargetLink Main Dialog Block](#) ([📖 TargetLink Model Element Reference](#)) is limited to generic RTOS and generic OSEK.
- Customizing which real-time operating systems are displayed in any user interface via the TargetLink Preferences Editor is no longer possible.

Additionally, the following restrictions apply to the CounterAlarm block:

- The CounterValue output must be connected to a Terminator block.
- The CounterTrigger input can be triggered only from outside the TargetLink subsystem.

The removal of the support of special OSEK versions was announced with TargetLink 4.1.

Related documentation

- [Connecting Inports and Outports of the CounterAlarm Block](#) ([📖 TargetLink Multirate Modeling Guide](#))

User state flags in Stateflow

With TargetLink 5.0, the support for the TargetLink-specific user state flags feature in Stateflow has been removed

As a replacement, you can use active state data with child activity mode. Refer to [Basics on Working with Active State Data](#) ([📖 TargetLink Preparation and Simulation Guide](#)).

The removal of the support of user state flags was announced with TargetLink 4.3.

Support of AUTOSAR 2.x and 3.x

With TargetLink 5.0, the support of AUTOSAR 2.x and AUTOSAR 3.x has been removed.

The removal of the support for AUTOSAR 2.x and AUTOSAR 3.x was announced with TargetLink 4.3.



TargetLink Blockset (stand-alone)

With TargetLink 5.0, the setup routine of the TargetLink Blockset (stand-alone) has been discontinued.

If you do not require the features of a fully-featured TargetLink installation, you can install TargetLink via the dSPACE Setup and adjust the operation mode to **Modeling Only**. You can adjust the operation mode via the TargetLink Preferences Editor or via the API command `tlOperationMode`.

The removal of the Blockset (stand-alone) setup was announced with TargetLink 4.4.

Related documentation

- [tlOperationMode](#) ( [TargetLink API Reference](#))
- [Overview of the TargetLink Operation Modes](#) ( [TargetLink Blockset Guide](#))

Obsolete API Functions

Obsolete API functions

Function	Status	Replacement
tl_adapt_dd_references	Error ¹⁾	tlMoveDDObject
tl_extract_subsystem	Error ¹⁾	tlExtractSubsystem
tl_find_dd_references	Error ¹⁾	tlFindDDReferences
tl_get_blockset_mode	Error ¹⁾	tlOperationMode
tl_switch_blockset	Error ¹⁾	tlOperationMode
tl_sim_interface	Error ¹⁾	tlSimInterface
tl_upgrade	Error ¹⁾	tlUpgrade
generate_ASAP2	Error ¹⁾	dsdd_export_a2l_file
tl_upgrade_libmapfile ²⁾	Error ¹⁾	-

¹⁾ The function was removed from TargetLink.

²⁾ The support for libmaps in the TargetLink Version 2.x format was removed.

Note

See the help contents on the new API functions to adjust your user scripts accordingly.

**Obsolete Data Dictionary
MATLAB API functions**

Function	Status	Replacement
CreateFunctionDocumentation ¹⁾	Error ²⁾	CreateDocumentation
GetFunctionDocumentation ¹⁾	Error ²⁾	GetDocumentation
GetFunctionDocumentationBlockComment ¹⁾	Error ²⁾	GetBlockComment
GetFunctionDocumentationCodeComment ¹⁾	Error ²⁾	GetCodeComment
SetFunctionDocumentationBlockComment ¹⁾	Error ²⁾	SetBlockComment
SetFunctionDocumentationCodeComment ¹⁾	Error ²⁾	SetCodeComment

¹⁾ With TargetLink 5.0, DD Documentation objects changed. For more information, refer to [DD Documentation objects](#) on page 191.

²⁾ The function was removed from TargetLink.

Obsolete Limitations

Obsolete with TargetLink 5.0	With TargetLink 5.0, the following limitations of previous TargetLink versions were removed:
Simulink.Bus objects	The use of <code>Simulink.Bus</code> objects in Stateflow diagrams is not supported. For Stateflow data that are <code>Simulink.Bus</code> objects, the following error message will be displayed: E20950 The data type Bus Object is currently unsupported for the variable <x> in state charts.
No export of CodeDecoration objects	For AUTOSAR version 2.1.4, TargetLink does not support the export of CodeDecoration objects.
DD API does not detect data model compliance of partial DD files	The DD API does not detect if a partial DD file complies with the current data model. For example, if you load a partial DD file created with a different TargetLink version, the active DD workspace might become invalid.

Changes in Future TargetLink Versions

Where to go from here

Information in this section

Features to Be Discontinued	211
API Functions to Be Discontinued	212
Deprecated Code Generator Options	212

Features to Be Discontinued

RTOS/OSEK code generation mode

Support for the TargetLink RTOS/OSEK Code Generation Modes will be discontinued in future TargetLink versions.

Target Optimizations Module (TOM)

Support for the TargetLink Target Optimizations Module (TOM) will be discontinued in future TargetLink versions.

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. This means the code differs only 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.

Simulink's classic initialization mode

Support for the Simulink classic initialization mode 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.

Automatic interpretation of Boolean

The automatic interpretation of certain integer data types as Boolean by TargetLink will be discontinued in future TargetLink versions.

Unit Delay Reset Enabled	The TargetLink support of the Unit Delay Reset Enabled block will be discontinued in future TargetLink versions.
Model change detection via checksum	The TargetLink support of model change detection via checksum will be discontinued in future TargetLink versions.
DD ModuleTemplate	DD ModuleTemplate objects 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_compare_fcn_signature	TargetLink 5.0	-

Note

See the help contents on the new API functions to adjust your user scripts accordingly.

Deprecated Code Generator Options

Deprecated Code Generator Options The following Code Generator options are deprecated and will be removed in future TargetLink versions:

- **AllowStructAssignments** ([📖 TargetLink Model Element Reference](#))

VEOS


Where to go from here


Information in this section

New Features of VEOS 4.5	213
Gives an overview of the new features of VEOS 4.5.	
Compatibility of VEOS 4.5	214
Provides information on the compatibility of VEOS 4.5.	
Migrating to VEOS 4.5	217
To migrate from VEOS 4.4 to VEOS 4.5, you might have to perform certain migration steps.	
Discontinuations in VEOS 4.5	218
Provides information on the features discontinued as of VEOS 4.5.	

New Features of VEOS 4.5

General enhancement

Support of the Microsoft Visual C/C++ Compiler from Microsoft Visual Studio 2017 VEOS 4.5 supports version 15 of the Microsoft Visual C/C++ Compiler (provided by Microsoft Visual Studio 2017 or Visual Studio 2017 Express). Refer to [Simulation Target Manager](#) ( [VEOS Manual](#)).

Specifying C/C++ compiler options separately VEOS 4.5 now lets you specify compiler options for C and C++ *separately*. Refer to [Import](#) ( [VEOS Manual](#)).

Enhancements for adaptive V-ECUs

Support of AP R19-03 VEOS 4.5 supports R19-03 of the AUTOSAR Adaptive Platform. Refer to [AUTOSAR Adaptive Platform](#) on page 217.

Enhancements for classic V-ECUs

Simulation of V-ECUs with LIN (slave) driver on VEOS VEOS now also supports the simulation of V-ECUs with the LIN driver MCAL module for LIN slaves. Refer to [Basic Software Module Support for V-ECUs](#) ([📖 Virtual Validation Overview](#)).

Sleep/wake-up simulation of V-ECUs on VEOS VEOS now supports the simulation of sleep/wake-up scenarios for V-ECUs.

Support of CAN RX interrupts VEOS now supports CAN RX interrupts in connection with the offline simulation of V-ECUs and bus VPUs.

Compatibility of VEOS 4.5

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](#)).

Supported operating systems

For information on the operating systems supported by VEOS, refer to [Operating System](#) on page 220.

Note

If you work with VEOS in Windows 7, make sure that the Microsoft Windows update KB2533623 is installed. This update is *not* part of Service Pack 1.

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	Version
... from VEOS 4.5	Real-Time Testing Version 4.2
... from VEOS 4.4	Real-Time Testing Version 4.1
... from VEOS 4.3	Real-Time Testing Version 4.0
... from VEOS 4.2	Real-Time Testing Version 3.4

VEOS Simulator	Version
... from VEOS 4.1	Real-Time Testing Version 3.3
... from VEOS 4.0	Real-Time Testing Version 3.2

ControlDesk 7.1 automatically uses the VEOS Simulator of VEOS 4.5. You can therefore use RTT in connection with VEOS and ControlDesk if RTT 4.2 is active on the PC.

File import compatibility

BSC compatibility VEOS 4.5 is compatible with bus simulation container (BSC) files created with the Bus Manager of dSPACE Release 2019-B (BSC version 1.7).

CTLGZ compatibility The following table shows the compatibility between VEOS 4.5 and CTLGZ files (V-ECU implementations):

V-ECU Implementations Created With...	V-ECU Implementation Version
dSPACE Release 2019-B: <ul style="list-style-type: none"> ▪ SystemDesk 5.4 ▪ TargetLink 5.0 	2.10
dSPACE Release 2019-A: <ul style="list-style-type: none"> ▪ SystemDesk 5.3 	2.9
dSPACE Release 2018-B: <ul style="list-style-type: none"> ▪ SystemDesk 5.2 ▪ TargetLink 4.4 	2.8
dSPACE Release 2018-A: <ul style="list-style-type: none"> ▪ SystemDesk 5.1 	2.7

FMU compatibility

- VEOS supports Functional Mock-up Units (FMUs) that comply with the FMI 2.0 standard.
- VEOS supports only the *FMI for Co-Simulation interface*, but not the FMI for Model Exchange interface.

For detailed and up-to-date compatibility information on the dSPACE FMI support, refer to:

<http://www.dspace.com/go/FMI-Compatibility>.

OSA compatibility The following table shows the compatibility between VEOS 4.5 and offline simulation application (OSA) files:

OSA Files Created with Products Of ...	OSA Version
dSPACE Release 2019-B	4.5 ¹⁾
dSPACE Release 2019-A	4.4 ²⁾

OSA Files Created with Products Of ...	OSA Version
dSPACE Release 2018-B	4.3 ²⁾
dSPACE Release 2018-A	4.2 ²⁾

- ¹⁾ OSA files created or modified with VEOS 4.5 cannot be loaded in earlier VEOS versions.
- ²⁾ 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.

SIC compatibility The following table shows the compatibility between VEOS 4.5 and Simulink implementation container (SIC) files:

SIC Files Created With ...	SIC Version
dSPACE Release 2019-B: <ul style="list-style-type: none"> ▪ Model Interface Package for Simulink 4.2 ▪ TargetLink 5.0 	1.7 ¹⁾
dSPACE Release 2019-A: <ul style="list-style-type: none"> ▪ Model Interface Package for Simulink 4.1 	1.6 ¹⁾
dSPACE Release 2018-B: <ul style="list-style-type: none"> ▪ Model Interface Package for Simulink 4.0 ▪ TargetLink 4.4 	1.5 ¹⁾
dSPACE Release 2018-A: <ul style="list-style-type: none"> ▪ Model Interface Package for Simulink 3.6 	1.4 ¹⁾

- ¹⁾ If the SIC file is created with a previous dSPACE Release and if the SIC file contains an ASM model, you cannot simulate the model in VEOS 4.5 (dSPACE Release 2019-B). For more information, refer to [Migrating ASM Models](#) (📖 VEOS Manual).

SMC compatibility The following table shows the compatibility between VEOS 4.5 and system model container (SMC) files:

SMC Files Created With ...	SMC Version
dSPACE Release 2019-B <ul style="list-style-type: none"> ▪ SYNECT 2.8 ▪ VEOS 4.5 	1.1
dSPACE Release 2019-A <ul style="list-style-type: none"> ▪ SYNECT 2.7 ▪ VEOS 4.4 	1.1
dSPACE Release 2018-B: <ul style="list-style-type: none"> ▪ SYNECT 2.6 ▪ VEOS 4.3 	1.1
dSPACE Release 2018-A: <ul style="list-style-type: none"> ▪ SYNECT 2.5 ▪ VEOS 4.2 	1.1

You also have to consider the following compatibility restrictions of the individual container files contained in the SMC file to be imported: If the SMC file contains a container of an unsupported version, VEOS 4.5 imports neither the

unsupported container nor the connections to the application process based on the unsupported container.

SMC files exported with VEOS 4.5 have file version 1.1.

Hypervisor compatibility

The simulation of adaptive V-ECUs with VEOS requires a hypervisor.

The following hypervisor software is supported:

- VMware Workstation 14 Player or Pro
- Microsoft Hyper-V on all Windows 10 operating systems supported by dSPACE. Refer to [Operating System](#) on page 220.

Only one hypervisor can be active on the Windows operating system. If several hypervisors are installed, VEOS uses the active one.

Hardware support for virtualization (Intel VT-X/AMD-V) in the BIOS or UEFI of the host PC must be activated.

AUTOSAR Adaptive Platform

VEOS 4.5 is compatible with the following release of the AUTOSAR Adaptive Platform:

- R19-03

Migrating to VEOS 4.5

Introduction

To migrate from VEOS 4.4 to VEOS 4.5, you might have to perform certain migration steps.

Note


To migrate to VEOS 4.5 from versions earlier than 4.4, you might also have to perform the migration steps of the intervening VEOS versions.

Migrating ASM models

You cannot simulate an ASM model on VEOS if the model is contained in an OSA or SIC file that was created with a dSPACE Release earlier than the one to which your VEOS installation belongs.

To simulate an ASM model that was last saved with a dSPACE Release earlier than the dSPACE Release to which your VEOS version belongs, perform the following steps:

1. Migrate the ASM model to the dSPACE Release to which your VEOS version belongs.

For information on migrating ASM models, refer to [Migrating ASM Models](#) ( [ASM User Guide](#)).

2. Generate a Simulink implementation container (SIC) file on the basis of the ASM model by using the *Model Interface Package for Simulink*.
For instructions, refer to [Generating Simulink Implementation Containers](#) ([📖 Model Interface Package for Simulink - Modeling Guide](#)).
3. Import the SIC file to the VEOS Player of your VEOS version.
For instructions, refer to [How to Import Simulink Implementations](#) ([📖 VEOS Manual](#)).

Migrating from prior VEOS versions

To migrate from previous 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 4.5

Discontinuations as of VEOS 4.5

Microsoft Visual C/C++ Compiler versions 11 and 14 As of version 4.5, VEOS no longer supports the following versions of the Microsoft Visual C/C++ Compiler:

- 11 (provided by Microsoft Visual Studio 2012 or Visual Studio 2012 Express)
- 14 (provided by Microsoft Visual Studio 2015 or Visual Studio 2015 Express)

As of version 4.5, VEOS supports version 15 of the Microsoft Visual C/C++ Compiler (provided by Microsoft Visual Studio 2017 or Visual Studio 2017 Express).

Refer to [Simulation Target Manager](#) ([📖 VEOS Manual](#)).

Related topics**Basics**

[Discontinuations in VEOS](#) ([📖 VEOS Manual](#))

Compatibility Information

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

Supported MATLAB Releases	219
Operating System	220
Using dSPACE Software on Virtual Machines (VMs)	223
Run-Time Compatibility of dSPACE Software	226
Limitations for Using Windows Features	227

Supported MATLAB Releases

MATLAB®/Simulink®

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 2019-B					
	RCP and HIL Software ^{1), 2)}	AutomationDesk 6.2 ³⁾	TargetLink 5.0	Model Compare 3.0	dSPACE Python Extensions 3.2 ⁴⁾	XIL API .NET MAPort 2019-B
R2019b	✓ ⁵⁾	✓	✓	✓	✓	✓
R2019a	✓	✓ ⁶⁾	✓	✓ ⁶⁾	✓ ⁷⁾	✓
R2018b	✓	✓	✓	✓	✓	✓
R2018a	✓	✓	✓	✓	✓	✓

- 1) 'RCP and HIL software' is a generic term for a software package containing several dSPACE software products, for example, ASM, RTI, ConfigurationDesk, MotionDesk and ModelDesk. These software products are installed in a common folder.
- 2) MATLAB/Simulink Student Suite does not support Automotive Simulation Models (ASM).
- 3) The AutomationDesk MATLAB Access Library requires MATLAB.
- 4) matlablib2 of dSPACE Python Extensions requires MATLAB.
- 5) R2019b is not supported by the RTI FPGA Programming Blockset – FPGA Interface.
- 6) R2019a is only supported by Model Compare and the MATLAB Access Library in AutomationDesk if at least R2019a Update 5 is used.
- 7) R2019a is only supported by the matlablib2 if at least R2019a Update 5 is used.

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>.

Operating System

Operating system on host PC

The dSPACE products of dSPACE Release 2019-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.

Note

dSPACE support of Windows 7 will end with dSPACE Release 2019-B (November 2019). Microsoft® is planning to end its support for Windows 7. The extended support will end on January 14, 2020. Thereafter, Microsoft will no longer provide security patches or new support information. Therefore, dSPACE Release 2019-B will be the final software version that will be released for Windows 7.

- The following editions, channels, 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
 - Long-Term Servicing Channel: LTSC 2019
 - Semi Annual Channel (formerly known as Current Branch (CB)): The compatibility statement of Microsoft applies. This means that newer versions released in this channel should be compatible with all previous versions. dSPACE used the 1809 version of the Semi Annual Channel 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 227.

Using MicroAutoBox Embedded PC as host PC

ControlDesk can also be installed on:

- MicroAutoBox Embedded PC 3rd Gen. Intel® Core™ i7-3517UE Processor, running on Windows 7 Professional, Ultimate, and Enterprise, 64-bit version
- MicroAutoBox Embedded PC 6th Gen. Intel® Core™ i7-6822EQ Processor, running on Windows 10 IOT Enterprise, LTSB 2016, 64-bit version

Operating system on SYNECT server

The SYNECT server supports the following operating systems:

- The same operating systems as listed above for all dSPACE products of dSPACE Release 2019-B.
- Windows Server 2016. The Windows Server Semi Annual Channel versions are not supported.

Note

Do not install the SYNECT client on a Windows server operating system, such as Windows Server 2016.

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 tests license servers only 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 described above.

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>\dSPACERCPHIL2019-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), 8090 (TCP), 9923 (UDP), 15000 (UDP), 49152 ... 65535 (TCP, TCP6 and UDP)
- dSPACE Installation Manager and CodeMeter licensing software require the following open TCP/IP network ports:
 - 22350 (TCP and UDP) for communication in a LAN network (if not changed from the default setting).
 - 22352 (TCP and UDP): To access CodeMeter WebAdmin via http.
 - 22353 (TCP and UDP): To access CodeMeter WebAdmin via https.

- dSPACE Help requires an open TCP/IP network port for interprocess communication between its components. The default port number is 11000. If this port number is already being used, another free port is used automatically. The related processes can be identified via the following prefixes: `HelpApsLayer<xxx>`, `HelpInstaller<xxx>`.

Using dSPACE Software on Virtual Machines (VMs)

Introduction

As of dSPACE Release 2019-A, you can operate several dSPACE products installed on virtual machines. However, some dSPACE products support VMs only with limitations, and other dSPACE products cannot be operated on VMs at all.

Usage restrictions

Note

The dSPACE End User License Agreement (EULA) does not allow:

- Using a virtual machine for circumventing license protection mechanisms, for multiple use of an acquired license or for use outside the use determined by the license type.
- Accessing dSPACE software via Internet or network applications (e.g., Citrix, Microsoft Remote Desktop or other terminal/device servers) or to grant such access to third parties.

If you have any questions or encounter any problems, contact dSPACE Support (www.dspace.com/go/supportrequest).

Recommended virtual machine software

dSPACE tests the functionality of dSPACE software products with current VMware products and VM hardware compatibility version 10 and version 13.

Support of dSPACE software on virtual machines

Note

The following table shows the compatibility for all dSPACE products. For products that support VMs with limitations, the known limitations are listed. For these products, further limitations might apply depending on the use case.

Product	Full Support	Support with Known Limitations	No Support
ASM	✓	—	—
AutomationDesk	—	✓ Known limitations: <ul style="list-style-type: none"> ▪ Access to DS1005/DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1005/DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible. 	—
Bus Manager	✓	—	—
ConfigurationDesk - Configuration Version	—	✓ Limitations apply if the RapidPro system is used as an intelligent I/O subsystem to extend DS1005 modular systems. ConfigurationDesk cannot access a real-time application if the DS1005 modular system is connected to the host PC via dSPACE link boards. If the DS1005 modular system is connected to the host PC via an Ethernet connection and a slot CPU, communication and therefore performance is very low.	—
ConfigurationDesk - Implementation Version	✓	—	—
Container Manager	✓	—	—
ControlDesk	—	✓ Known limitations: <ul style="list-style-type: none"> ▪ Access to DS1005/DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1005/DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible. 	—
Data Dictionary Manager	✓	—	—
DCI-GSI Configuration Package	✓	—	—
dSPACE Installation Manager	✓	—	—
ECU Flash Programming Tool	✓	—	—
ECU Interface Base Package	✓	—	—
ECU bypassing target compiler	✓	—	—
Failure Simulation Package	—	✓ Supported only in combination with the VEOS platform. Combinations with other platforms are not tested and therefore not released for use on VMs.	—
Firmware Archives	—	✓ Known limitations: <ul style="list-style-type: none"> ▪ Access to DS1005/DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1005/DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible. 	—
Firmware Manager			
FlexRay Configuration Tool	✓	—	—
Model Compare	✓	—	—

Product	Full Support	Support with Known Limitations	No Support
ModelDesk	—	✓ Known limitations: <ul style="list-style-type: none"> ▪ The Traffic Object Manager cannot show custom sensor points in the preview. ▪ Plotting occasionally does not start if a start trigger is used. 	—
Model Interface Package for Simulink	✓	—	—
MotionDesk	—	—	✓ ¹⁾
Platform API Package	—	✓ Known limitations: <ul style="list-style-type: none"> ▪ Access to DS1005/DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1005/DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible. 	—
Real-Time Testing	—	✓ Known limitations: <ul style="list-style-type: none"> ▪ Access to DS1005/DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1005/DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible. 	—
RTI Blocksets (Real-Time Interface)	—	✓ Known limitations: <ul style="list-style-type: none"> ▪ Access to DS1005/DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1005/DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore the performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible. 	—
Sensor Simulation	—	—	✓ ¹⁾
SYNECT	✓	—	—
SYNECT Server	✓	—	—
SYNECT License Server	✓	—	—
SystemDesk	✓	—	—
TargetLink	✓	—	—
VEOS	✓ ²⁾	—	—

¹⁾ VMs do not fulfill the requirements for graphics adapters.

²⁾ If you want to simulate adaptive AUTOSAR V-ECUs and use Hyper-V (Hypervisor from Windows 10), the VM hardware compatibility version 12 or higher is required.

Required knowledge for setting up a virtual machine

To set up a virtual machine, you must have knowledge about the technology of VMs.

In virtual environments, significantly higher latencies and lower network performance (network throughput) must be expected compared to physical PCs. dSPACE has no influence on this.

Using virtual machines in parallel If you use multiple VMs simultaneously on one PC, sharing of host resources such as CPU, network and disk I/O bandwidth can cause timing issues. dSPACE recommends to use a physical PC if high performance is required by an application.

Using a virtual machine on the host PC

System requirements PCs that host virtual machines with dSPACE software, must meet at least the requirements listed in [Appendix: System Requirements](#) ([📖 Installing dSPACE Software](#)). You are recommended to use a PC with more resources so that the software runs smoothly on a VM, because the VM software itself uses up some of the resources:

- The CPU speed and RAM size must be sufficient to run the operating system and the software on the host PC as well as the guest operating system and the application software on the VM.
- You also require sufficient free disk space to install the VM software and the software you want to run, just as you would if you were installing it directly on your PC.

Connecting dongle-based devices If you use dongle-based single-user licenses to use dSPACE software, you first have to connect your CmDongle to the host PC. Then you have to connect the WIBU-Systems CodeMeter-Stick device to the virtual machine on the host PC. For instructions, refer to the documentation of the VM software you use.

Using floating network licenses If you use floating network licences, the virtual machine requires access to the dSPACE License Server. For further instructions, refer to [How to Set up a Connection Between Client and Server](#) ([📖 Working with CodeMeter Licensing Technology](#)).

Optimal display of dSPACE Help For an optimal display of the content in dSPACE Help, you have to activate the ClearType setting in the VM (= default setting).

You can access this setting via the Windows Start menu (Start – Control Panel – Appearance and Personalization – Display – Adjust ClearType text).

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 2019-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 used together.

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 required, refer to <http://www.dspace.com/go/CompPatch>.

- RCP and HIL software products (of Release 2019-B) cannot be used in combination with RCP and HIL software products from earlier dSPACE Releases.

Major limitation for working with a SCALEXIO system and with MicroAutoBox III

The products for working with a SCALEXIO system and with MicroAutoBox III must be compatible. This is guaranteed only for products delivered with the same dSPACE Release. Contact dSPACE for more information.

Compatibility of real-time applications loaded to a DS1005, DS1006, DS1104 or MicroAutoBox II 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.

Limitations for Using Windows Features

Motivation

Some limitations apply to 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.

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). For more information on FIPS, refer to <https://technet.microsoft.com/en-us/library/security/cc750357.aspx>.

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.

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.

This affects all MATLAB versions and all Windows operating systems, that are supported by dSPACE.

**Valid for Windows 10:
Microsoft .NET Framework 3.5
feature must be enabled**

The Microsoft .NET Framework 3.5 feature must be installed and enabled. This is required for using client programs and libraries built with .NET Runtime 2.0. If the Microsoft .NET Framework 3.5 is not enabled, the dSPACE software installation is interrupted and an error message is displayed.

With dSPACE Release 2020-A, dSPACE will discontinue the support for client programs and libraries built with .NET Runtime 2.0. This applies to any C#-based programs using the provided interfaces (APIs) for COM automation and for dSPACE Installation Manager. Applications using these interfaces have to use at least .NET Runtime 4.0.

A

ASM Base InCylinder Blockset
 migration 34
 new features 34

ASM Brake Hydraulics Blockset
 migration 36

ASM Diesel Engine Blockset
 new features 37

ASM Diesel Exhaust Blockset
 migration 38
 new features 38

ASM Diesel InCylinder Blockset
 new features 40

ASM Drivetrain Basic Blockset
 migration 41

ASM Electric Components Blockset
 migration 43
 new features 42

ASM Engine Gasoline Basic Blockset
 new features 46

ASM Engine Gasoline Blockset
 new features 47

ASM Environment Blockset
 migration 44
 new features 44

ASM Gasoline InCylinder Blockset
 new features 48

ASM Traffic Blockset
 migration 51
 new features 49

ASM Vehicle Dynamics Blockset
 new features 55

AutomationDesk
 migration 31
 new features 29

B

Bus Manager (stand-alone)
 migration 59
 new features 57

C

Common Program Data folder 12

CommonProgramDataFolder 12

ControlDesk
 migration 84
 new features 78

D

discontinuation
 hardware 17
 planned 17

Discontinuation 17

Documents folder 12

DocumentsFolder 12

dSPACE FlexRay Configuration Package
 new features 91

dSPACE Python Extensions

new features 93

dSPACE XIL API
 new features 99

E

ECU Interface Manager
 migration 103
 new features 101

F

Firmware Manager
 new features 105

G

general enhancements and changes 15

H

host PC software
 MATLAB 219
 operating system 220

K

key features 20

L

Limitations
 TargetLink
 obsolete limitations 210

limitations for using Windows features 227

Local Program Data folder 12

LocalProgramDataFolder 12

M

MATLAB
 requirements 219
 supported releases 219

MATLAB limitations (R2019b)
 RTI/RTI-MP 122

migration
 ASM Base InCylinder Blockset 34
 ASM Brake Hydraulics Blockset 36
 ASM Diesel Exhaust Blockset 38
 ASM Drivetrain Basic Blockset 41
 ASM Electric Components Blockset 43
 ASM Environment Blockset 44
 ASM Traffic Blockset 51
 AutomationDesk 31
 Bus Manager (stand-alone) 59
 ControlDesk 84
 ECU Interface Manager 103
 Model Compare 109
 ModelDesk 112
 MotionDesk 117
 Real-Time Testing 119
 RTI 122
 RTI Bypass Blockset 125
 RTI CAN MultiMessage Blockset 128

RTI FPGA Programming Blockset 131

RTI LIN MultiMessage Blockset 133

Model Compare
 migration 109
 new features 107

ModelDesk
 migration 112
 new features 111

MotionDesk
 migration 117
 new features 117

N

new features
 ASM Base InCylinder Blockset 34
 ASM Diesel Engine Blockset 37
 ASM Diesel Exhaust Blockset 38
 ASM Diesel InCylinder Blockset 40
 ASM Electric Components Blockset 42
 ASM Engine Gasoline Basic Blockset 46
 ASM Engine Gasoline Blockset 47
 ASM Environment Blockset 44
 ASM Gasoline InCylinder Blockset 48
 ASM Traffic Blockset 49
 ASM Vehicle Dynamics Blockset 55
 AutomationDesk 29
 Bus Manager (stand-alone) 57
 ControlDesk 78
 dSPACE FlexRay Configuration Package 91
 dSPACE Python Extensions 93
 dSPACE XIL API 99
 ECU Interface Manager 101
 Firmware Manager 105
 Model Compare 107
 ModelDesk 111
 MotionDesk 117
 Real-Time Testing 119
 RTI Bypass Blockset 125
 RTI CAN MultiMessage Blockset 127
 RTI FPGA Programming Blockset 129
 RTI Synchronized Time Base Manager
 Blockset 135
 RTI/RTI-MP 121
 RTLib 121
 SCALEXIO firmware 137
 SystemDesk 152
 test management 141
 VEOS 213
 workflow management 143

new general features
 SYNECT 140

new hardware 15

P

planned discontinuation
 software support 17

product overview 18

Python distribution 16

R

Real-Time Testing
 migration 119
 new features 119
 requirements
 host PC software
 MATLAB 219
 operating system 220
 RTI Bypass Blockset
 migration 125
 new features 125
 RTI CAN MultiMessage Blockset
 migration 128
 new features 127
 RTI FPGA Programming Blockset
 migration 131
 new features 129
 RTI LIN MultiMessage Blockset
 migration 133
 RTI Synchronized Time Base Manager Blockset
 new features 135
 RTI/RTI-MP
 MATLAB limitations (R2019b) 122
 new features 121
 RTLib
 new features 121

S

SCALEXIO firmware
 new features 137
 supported MATLAB releases 219
 SYNECT
 new general features 140
 system requirements
 operating system 220
 SystemDesk
 new features 152

T

TargetLink
 AUTOSAR features, new
 supported releases 162
 code generator options
 backward compatibility 184
 changed default value 184
 discontinued features 208
 migrating to
 new version 175
 migration
 obsolete limitations 210
 new API functions 171
 new code generator options 170
 new features 158
 general changes 173
 general enhancements 173
 new version
 migrating to 175
 target support
 discontinued compiler versions 166
 discontinued evaluation boards 166

new compiler versions 166
 new evaluation boards 166
 supported targets 166
 TargetLink Data Dictionary
 migrating to
 new version 175
 migration 175
 discontinued documentation 175
 manually upgrading libraries and models
 179
 upgrading existing data dictionaries 178
 new version
 migrating to 175
 test management
 new features 141

V

VEOS
 new features 213
 version history 18

W

Windows
 limitations 227
 workflow management
 new features 143