

dSPACE Release

New Features and Migration

Release 2023-B – November 2023

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
E-mail:	info@dspace.de
Web:	https://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 have 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 the software updates and patches themselves and for more information, such as how to receive an automatic notification when an update or a patch is available for your dSPACE software.

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 - 2023 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

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

AURELION, AUTERA, ConfigurationDesk, ControlDesk, MicroAutoBox, MicroLabBox, SCALEXIO, SIMPHERA, 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
Overview of dSPACE Release 2023-B	13
General Enhancements and Changes.....	13
Discontinuations.....	15
Product Version Overview.....	17
New Key Product Features.....	20
New Use Scenario: Data Replay.....	25
Aspects of Migrating from Previous Releases	27
Migrating to dSPACE Release 2023-B.....	27
Migrating Python Scripts from Python 3.9 to Python 3.11	29
Main Changes in Python 3.11.....	29
Main Changes in Handling Python 3.11 with dSPACE Software.....	30
General Information on Using Python Installations.....	31
Technical Changes.....	33
Product-Specific Information on Migrating.....	33
AutomationDesk	37
New Features of AutomationDesk 2023-B.....	37
Migrating to AutomationDesk 2023-B.....	38
Automotive Simulation Models (ASM)	39
All ASM Products.....	40
Changes in All ASM Demo Models.....	40
Migrating All ASM Blocksets.....	40
ASM Battery.....	41
New Features of ASM Battery Blockset 2023-B.....	41
Changes in the ASM Battery Demo Models.....	41
Migrating to ASM Battery Blockset 2023-B.....	42
ASM Diesel Engine.....	43
Migrating to ASM Diesel Engine Blockset 2023-B.....	43

ASM Drivetrain Basic.....	44
New Features of ASM Drivetrain Basic Blockset 2023-B.....	44
Migrating to ASM Drivetrain Basic Blockset 2023-B.....	44
ASM Electric Components.....	45
New Features of ASM Electric Components Blockset 2023-B.....	45
Changes in the ASM Electric Components Demo Models.....	45
Migrating to ASM Electric Components Blockset 2023-B.....	45
ASM Fuel Cell.....	46
New Features of ASM Fuel Cell Blockset 2023-B.....	46
Changes in the ASM Fuel Cell Demo Model.....	47
Migrating to ASM Fuel Cell Blockset 2023-B.....	47
ASM Gasoline Engine.....	48
Migrating to ASM Gasoline Engine Blockset 2023-B.....	48
ASM Traffic.....	49
Changes in the ASM Traffic Demo Model.....	49
ASM Trailer.....	50
New Features of ASM Trailer Blockset 2023-B.....	50
Changes in the ASM Trailer Demo Model.....	51
ASM Truck.....	52
New Features of ASM Truck Blockset 2023-B.....	52
Changes in the ASM Truck Demo Model.....	52
ASM Vehicle Dynamics.....	54
New Features of ASM Vehicle Dynamics Blockset 2023-B.....	54
Changes in the ASM Vehicle Dynamics Demo Model.....	54
Migrating to ASM Vehicle Dynamics Blockset 2023-B.....	55
Bus Manager (Stand-Alone)	57
New Features of the Bus Manager (Stand-Alone) 2023-B.....	57
Migrating to the Bus Manager (Stand-Alone) 2023-B.....	60
Migrating Projects and Applications Created with Previous Versions.....	60
Bus Manager (Stand-Alone) Discontinuations.....	63
ConfigurationDesk	65
New General Features of ConfigurationDesk 2023-B.....	65
New Features Concerning I/O Functionality and Hardware Support.....	67
New Features of the Bus Manager in ConfigurationDesk.....	71
Supported Container File Versions.....	74
Migrating to ConfigurationDesk 2023-B.....	77

Migrating Projects and Applications Created with Previous Versions.....	80
ConfigurationDesk Discontinuations.....	83
ControlDesk	85
New Features of ControlDesk 2023-B.....	86
New Features of Platform Management and Platforms/Devices (ControlDesk 2023-B).....	86
New Variable Management Features (ControlDesk 2023-B).....	87
New Instrument Features (ControlDesk 2023-B).....	88
New Bus Navigator Features (ControlDesk 2023-B).....	91
New Electrical Error Simulation Features (ControlDesk 2023-B).....	99
New User Interface Handling Features (ControlDesk 2023-B).....	99
Migrating to ControlDesk 2023-B.....	102
Migrating to ControlDesk 2023-B.....	102
DCI Configuration Tool	107
New Features of the DCI Configuration Tool 2023-B.....	107
dSPACE ECU Flash Programming Tool	109
New Features of the dSPACE ECU Flash Programming Tool 2023-B.....	109
dSPACE FlexRay Configuration Package	111
New Features of dSPACE FlexRay Configuration Package 2023-B.....	111
dSPACE Installation Manager	113
New Features of dSPACE Installation Manager 23.2.....	113
Migrating to dSPACE Installation Manager 23.2.....	114
dSPACE Python Extensions	115
New Features of dSPACE Python Extensions 2023-B.....	115
dSPACE XIL API .NET	117
New Features of dSPACE XIL API .NET 2023-B.....	117
Migrating to dSPACE XIL API .NET 2023-B.....	118
ECU Interface Manager	121
New Features of ECU Interface Manager 2023-B.....	121

Compatibility of ECU Interface Manager 2023-B.....	122
Migrating to ECU Interface Manager 2023-B.....	122
Firmware Manager	125
New Features of Firmware Manager 23.2.....	125
FPGA Programming Blockset	127
New Features of the FPGA Programming Blockset 2023-B.....	127
Migrating to the FPGA Programming Blockset 2023-B.....	129
MicroAutoBox III Firmware	133
New Features of the MicroAutoBox III Firmware 23.2.....	133
Model Compare	135
New Features of Model Compare 2023-B.....	135
Migration to Model Compare 2023-B.....	136
Model and Sensor Interface Blockset	137
New Features of the Model and Sensor Interface Blockset 2023-B.....	137
Migrating to Model and Sensor Interface Blockset 2023-B.....	137
ModelDesk	139
New Features of ModelDesk 2023-B.....	139
Migration to ModelDesk 2023-B.....	140
Model Interface Package for Simulink	141
New Features of the Model Interface Package for Simulink 2023-B.....	141
Migrating to the Model Interface Package for Simulink 2023-B.....	142
MotionDesk	143
New Features of MotionDesk 2023-B.....	143
Migrating to MotionDesk 2023-B.....	143
Real-Time Testing	145
New Features of Real-Time Testing 2023-B.....	145
Migrating to Real-Time Testing 2023-B.....	145

RTI/RTI-MP and RTLib	147
New Features of RTI/RTI-MP and RTLib.....	147
Migration Aspects of RTI/RTI-MP and RTLib.....	147
RTI Bypass Blockset	149
Migrating to RTI Bypass Blockset 2023-B.....	149
RTI CAN MultiMessage Blockset	151
New Features of the RTI CAN MultiMessage Blockset 2023-B.....	151
Migrating to RTI CAN MultiMessage Blockset 2023-B.....	151
RTI LIN MultiMessage Blockset	153
New Features of the RTI LIN MultiMessage Blockset 2023-B.....	153
Migrating to RTI LIN MultiMessage Blockset 2023-B.....	153
RTI Synchronized Time Base Manager Blockset	155
New Features of the RTI Synchronized Time Base Manager Blockset 2023-B.....	155
SCALEXIO Firmware	157
New Features of the SCALEXIO Firmware 23.2.....	157
Migrating to SCALEXIO Firmware 23.2.....	158
Discontinuations in SCALEXIO Firmware.....	159
Signal List Editor	161
Features of Signal List Editor 2023-B (23.1).....	161
SYNECT	163
Migrating to SYNECT 2023-B.....	164
Migrating from SYNECT 2023-A.....	164
Migrating Databases.....	164
Data Model Changes from SYNECT 2023-A to SYNECT 2023-B.....	165

SystemDesk 167

- New Features of SystemDesk 2023-B..... 168
 - New General Features..... 168
 - New V-ECU Generation Features..... 169
 - Improved A2L File Handling..... 169
- Migrating to SystemDesk 2023-B..... 172
 - Discontinuations of dSPACE Release 2023-B..... 172
 - Migrating to SystemDesk 2023-B..... 173
 - Migrating API Scripts..... 174

TargetLink 179

- New Features of TargetLink 2023-B..... 180
 - Modeling in Simulink or Stateflow..... 180
 - Improved Array-of-Struct Support..... 181
 - Dynamic Element Specification..... 181
 - Solving Systems of Linear Equations 182
 - Support of Ensure Outport is Virtual..... 183
 - Support of Variable Vector Widths for Prelookup Blocks..... 183
 - Working with Stateflow Objects..... 184
 - Improvements Concerning Function Interfaces..... 184
 - Adaptive AUTOSAR..... 185
 - Supported Adaptive AUTOSAR Releases..... 185
 - Adaptive AUTOSAR Frame Model Generation 185
 - C++ Enumeration Class Data Types..... 186
 - New Adaptive AUTOSAR Modeling Features..... 186
 - Changes According to AUTOSAR Version R22-11..... 187
 - C++ Namespaces..... 187
 - Classic AUTOSAR..... 188
 - Supported Classic AUTOSAR Releases..... 188
 - Improved Array-of-Struct Support (Classic AUTOSAR)..... 188
 - AUTOSAR Import and Export Improvements..... 188
 - Code Generation Core Functionality..... 189
 - Improved Code Efficiency..... 189
 - Model-in-the-Loop Simulation (MIL)..... 190
 - Analyzing Simulation Results..... 190
 - Target Simulation (PIL)..... 190
 - Changes in the Target Simulation Modules with TargetLink 2023-B..... 190

Usability.....	191
Usability Improvements.....	191
Code Generator Options.....	194
New Code Generator Options.....	194
API Functions and Hook Scripts.....	195
New API Functions.....	195
New Hook Scripts.....	195
Other.....	196
FMI 3.0 Support.....	196
TargetLink Demos and Examples.....	197
Migrating to TargetLink 2023-B and TargetLink Data Dictionary 2023-B.....	199
General Migration Information.....	199
Upgrading Models, Libraries, and Data Dictionaries.....	199
Basics on Migrating Between TargetLink Versions.....	200
How to Upgrade and Save a Data Dictionary with Included DD Files.....	202
How to Manually Upgrade Libraries and Models Using the API.....	203
Migrating Data Dictionaries to CodeDecorationSets.....	204
Changes in Rounding Behavior for Exactly Representable Floating-Point Numbers (Microsoft).....	207
Changes in Rounding Behavior for Exactly Representable Floating-Point Numbers (Microsoft).....	207
Migrating from TargetLink 2022-B to 2023-B.....	207
Code Generator Options.....	208
Migration Aspects Regarding Code Generator Options.....	208
API Functions and Hook Scripts.....	210
Changes in API Functions.....	210
Changes in Hook Scripts.....	214
Other Migration Aspects.....	215
Various Migration Aspects.....	215
MATLAB Support.....	216
No Support of MATLAB R2023a and R2023b.....	216
Code Changes Between TargetLink 2022-B and TargetLink 2023-B.....	217
Major or Breaking Changes for Atomic Subsystems and Function Interfaces.....	217
Code Efficiency.....	222
Adaptive-AUTOSAR-Related and CPP-Related.....	230
Struct-Related and Array-of-Struct-Related.....	232

Classic-AUTOSAR-Related.....	235
Other Code Changes.....	237
Discontinuations as of TargetLink 2023-B.....	245
Discontinued TargetLink Features.....	246
Obsolete Limitations.....	246
Obsolete Code Generator Options.....	247
Changes in Future TargetLink Versions.....	248
Features to Be Discontinued.....	248
API Functions to Be Discontinued.....	249
Deprecated Code Generator Options.....	249
VEOS	251
New Features of VEOS 2023-B.....	251
Compatibility of VEOS 2023-B.....	253
Migrating to VEOS 2023-B.....	257
Discontinuations Planned for VEOS 2024-A.....	258
Compatibility Information	259
Supported MATLAB Releases.....	259
Operating System.....	261
Using dSPACE Software on Virtual Machines (VMs).....	263
Run-Time Compatibility of dSPACE Software.....	268
Limitations for Using Windows Features.....	268
Limitations for Using Linux Features.....	270
Index	271

About This Document

Content





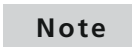



This document informs you about the new features of all the dSPACE software products in Release 2023-B. It also gives you an overview of software products with no or minor changes. There are instructions on migrating from earlier dSPACE Releases, especially from earlier product versions, if required.

Printed document

A printed copy of this document is available on demand. You can order it free of charge by using the following link:
<https://www.dspace.com/go/requestreleasematerial>.

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.
	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.
	Follows 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 Windows folders

Windows-based software products use the following special folders:

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

`%PROGRAMDATA%\dSPACE\`

or

`%PROGRAMDATA%\dSPACE\`

Documents folder A standard folder for application-specific files that are used by the current user.

`%USERPROFILE%\Documents\dSPACE\`

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


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

Accessing dSPACE Help and PDF Files

After you install and decrypt Windows-based dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

PDF files You can access PDF files via the  icon in dSPACE Help. The PDF opens on the first page.

dSPACE Help (Web) Independently of the software installation, you can access the Web version of dSPACE Help at <https://www.dspace.com/go/help>.

To access the Web version, you must have a *mydSPACE* account.

For more information on the mydSPACE registration process, refer to www.dspace.com/faq?097.

Overview of dSPACE Release 2023-B

Introduction Gives you an overview of the new key features in Release 2023-B and information about unchanged products.

Where to go from here

Information in this section

General Enhancements and Changes.....	13
Discontinuations.....	15
Product Version Overview.....	17
New Key Product Features.....	20
New Use Scenario: Data Replay.....	25

General Enhancements and Changes

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

Providing legal notes for using third-party software

Several dSPACE software products use or contain third-party software. The third-party components may be subject to additional terms and conditions or terms and conditions that deviate from the ones that apply to the dSPACE software. Copyright notices and license terms of third-party components must be adhered to. If the user is required to be explicitly informed of and to comply with the terms and conditions of third-party components, the relevant information on terms and conditions is stated in separate OSSAcknowledgements files for each dSPACE product.

You can find the OSSAcknowledgements files as follows:

- In the Legal folders accessible via the root directory on both dSPACE DVDs and the root directory of the downloaded installation files (mounted ISO images or unpacked ZIP files).
- After you install the dSPACE software in the <main installation path of dSPACE Setup>\<subfolder of InstallationSet>\Legal folder.

Python distribution

The support of Python 3.9 is discontinued with dSPACE Release 2023-A and is replaced by Python 3.11. For more information, refer to [Migrating Python Scripts from Python 3.9 to Python 3.11](#) on page 29.

Support of new dSPACE hardware

dSPACE Release 2023-B introduces new dSPACE hardware:

- MicroLabBox II

The new MicroLabBox II is the successor of the MicroLabBox, a ready-to-use rapid control prototyping system and hardware-in-the-loop test system for the laboratory environment:

- Improved computing power of the real-time processor and the real-time processor now provides four CPU cores.
- Improved communication interfaces, such as 10 GBit Ethernet ports and CAN FD SIC.
- Proven panel and connection concept: You can order a top panel variant with BNC connectors for analog signals or a front panel variant with Sub-D connectors.

New: You can stack up to three front panel variants.

- Supported by ConfigurationDesk to implement real-time applications.

For more information on the supported I/O functionality, refer to [Overview of MicroLabBox II I/O Functionality \(MicroLabBox II Hardware Installation and Configuration\)](#).

For getting started with MicroLabBox II, refer to [Get the MicroLabBox II Started \(MicroLabBox II - Getting Started\)](#).

For information for MicroLabBox and RTI users, refer to [Migrating MicroLabBox RTI Models \(MicroLabBox II - Getting Started\)](#).

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 for MATLAB R2023b:

- MinGW (GNU Compiler Collection (GCC 8.1.0))
- Microsoft Visual Studio 2022 Professional

Providing license overviews as PDF files

As of July 2023, dSPACE provides the license overviews (containing ticket IDs for the purchased products) as PDF files and no longer as Excel sheets.

New compliance check for run-time version licenses

Real-time applications often contain models that need run-time version licenses, for example, the run-time version licenses for Automotive Simulation Models

(ASM_..._RTV licenses). These licenses are required for downloading and executing a real-time application on the dSPACE real-time hardware, such as SCALEXIO.

As of dSPACE Release 2023-A, the dSPACE software performs a compliance check for such run-time version licenses when you download the real-time application. This check gives you the possibility to verify and fulfill the agreements in the dSPACE End User License Agreement (EULA).

Note

- With dSPACE Release 2023-A, dSPACE introduces the compliance check only for the required run-time version licenses for Automotive Simulations Models (ASM), for example, for the ASM Battery Library (run-time version) license. The check for run-time version licenses for other products will be included in later dSPACE Releases.
- Run-time version licenses for Automotive Simulations Models (ASM) are also required, if you want to execute offline simulation applications (OSAs) on the VEOS platform. An availability check for licenses required for OSAs has already been implemented since dSPACE Release 2017-B. This check is performed by VEOS when starting the simulation. If at least one of the required license is not available, executing the OSA is not possible.

For details, refer to [Compliance Check for Run-Time Version Licenses \(Working with CodeMeter Licensing Technology\)](#)

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 <https://www.dspace.com/go/discontinuation>.

Discontinuation of software support

SCALEXIO firmware As of dSPACE Release 2023-B, the 64-bit Linux-based real-time operating system (RTOS) is the default RTOS for a SCALEXIO system and the DS6001 Processor Board. The 32-bit Linux-based RTOS support was discontinued after dSPACE Release 2023-A. You can no longer generate 32-bit-based real-time applications.

Planned discontinuation of software support

PHS-bus hardware dSPACE Release 2023-B is the last Release that supports the PHS-bus hardware, i.e., the DS1006 Processor Board, the DS1007 PPC Processor Board, and all the PHS-bus I/O boards. This also applies to the dSPACE Simulator Mid-Size and the dSPACE Simulator EcoLine.

For new projects, we recommend that you use SCALEXIO as a modular real-time system.

MotionDesk MotionDesk and the MotionDesk Blockset will be discontinued after Release 2023-B. Until discontinuation, MotionDesk supports the simulation of camera sensors. Radar and lidar sensors are no longer supported using MotionDesk. For more information, refer to <https://www.dspace.com/go/elamd>.

RTI CAN/LIN MultiMessage Blockset for SCALEXIO RTI CAN MultiMessage Blockset and RTI LIN MultiMessage Blockset for SCALEXIO are planned for discontinuation after Release 2024-B. For more information, refer to <https://www.dspace.com/go/elarTlxxMMBS>.

For new SCALEXIO projects, we recommend that you use the Bus Manager.

Planned discontinuation of dSPACE hardware

MicroAutoBox II MicroAutoBox II with all its variants (1401/1507, 1401/1511, 1401/1511/1514, 1401/1513, and 1401/1513/1514) will be discontinued at the end of 2027. You can buy the product until the end of 2024 and the software support will be continued up to dSPACE Release 2026-B. For new projects, we recommend that you use MicroAutoBox III.

Discontinuation of legacy licensing technologies

As of dSPACE Release 2023-A, dSPACE Installation Manager no longer supports legacy licensing technologies (WibuKey from Wibu-Systems for dongle licenses and FlexNet from Flexera for floating network licenses). This means:

- dSPACE Installation Manager 23.1 (on dSPACE Release 2023-A) and later will support only the CodeMeter licensing technology introduced with dSPACE Release 2017-B.
- dSPACE Installation Manager 23.1 and later no longer provides the file download (keys.dsp and license.dsp) for legacy licensing based on CodeMeter licenses.
- Installing dSPACE Release 2023-A or later on the same PC as dSPACE Release 2017-A or earlier is no longer supported and therefore not recommended.
- As of dSPACE Release 2023-A, dSPACE License Manager (Legacy) is no longer shipped and installed with the dSPACE Installation Manager. The associated command line utilities are also discontinued, including in particular the **IMLicUtil.exe** and the **InstallationReporter.exe** utilities.
- If you still need to use the legacy licensing method, you must use the latest version of dSPACE Installation Manager that supports this technology. This is dSPACE Installation Manager 22.3. You can download dSPACE Installation Manager 22.3 from <https://www.dspace.com/go/IM>.

Discontinuation of the dongle migration with dSPACE Installation Manager

As of dSPACE Release 2023-A, dSPACE Installation Manager no longer supports migration from old dongles (delivered for dSPACE Release 2017-A and earlier) to currently delivered CmDongles.

If you still have an old dongle that needs to be migrated to use dSPACE Release 2017-B or later, the following applies:

- You are still using a WibuKey dongle (Type 1): Migration is no longer possible. A dongle exchange by dSPACE is required. For more information, refer to <http://www.dspace.com/go/DongleReplacement>.

- You are still using a first-generation CmDongle (Type 2), with serial number starting with 2-xxxx and white connector: Migration is no longer possible. A dongle exchange by dSPACE is required. For more information, refer to <http://www.dspace.com/go/DongleReplacement>.
- You are still using a CmDongle (Type 3), with serial number starting with 3-xxxx and old firmcode: Migration is required by using an earlier version of dSPACE Installation Manager (from 5.0 ... to 22.3). For the download of dSPACE Installation Manager 22.3, refer to <https://www.dspace.com/go/IM>.

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			
	2022-A	2022-B ¹⁾	2023-A	2023-B
AutomationDesk	6.7	2022-B	2023-A	2023-B Refer to AutomationDesk on page 37.
Automotive Simulation Models	10.1	2022-B	2023-A	2023-B Refer to Automotive Simulation Models (ASM) on page 39.
Bus Manager (stand-alone)	22.1	2022-B	2023-A	2023-B Refer to Bus Manager (Stand-Alone) on page 57.
ConfigurationDesk for RapidPro	– ²⁾	–	–	–
ConfigurationDesk	22.1	2022-B	2023-A	2023-B Refer to ConfigurationDesk on page 65.
Container Manager	5.2	–	–	–
ControlDesk	7.6	2022-B	2023-A	2023-B Refer to ControlDesk on page 85.
DCI Configuration Tool	3.13	2022-B	2023-A	2023-B Refer to DCI Configuration Tool on page 107.
dSPACE AUTOSAR Compare	1.1	2022-B	2022-B	2022-B
dSPACE Bus API Package	–	–	2023-A	2023-B
dSPACE CAN API Package	4.0.8	2022-B	2023-A	2023-B
dSPACE ECU Flash Programming Tool	2.8	2022-B	2023-A	2023-B Refer to dSPACE ECU Flash Programming Tool on page 109.
dSPACE FlexRay Configuration Package	4.9	2022-B	2023-A	2023-B Refer to dSPACE FlexRay Configuration Package on page 111.

Product	dSPACE Release			
	2022-A	2022-B ¹⁾	2023-A	2023-B
dSPACE Installation Manager	5.9	22.2	23.1	23.2 Refer to dSPACE Installation Manager on page 113.
dSPACE Python Extensions	4.2	2022-B	2023-A	2023-B Refer to dSPACE Python Extensions on page 115.
dSPACE XIL API .NET	2022-A	2022-B	2023-A	2023-B Refer to dSPACE XIL API .NET on page 117.
ECU Interface Manager	2.11	2022-B	2023-A	2023-B Refer to ECU Interface Manager on page 121.
Firmware Manager	3.5	22.2	23.1	23.2 Refer to Firmware Manager on page 125.
FPGA Programming Blockset ³⁾	3.13	2022-B	2023-A	2023-B Refer to FPGA Programming Blockset on page 127.
MicroAutoBox III firmware	6.0	22.2	23.1	23.2 Refer to MicroAutoBox III Firmware on page 133.
Model and Sensor Interface Blockset	1.3	2022-B	2023-A	2023-B Refer to Model and Sensor Interface Blockset on page 137.
Model Compare	3.2	2022-B	2022-B	2023-B Refer to Model Compare on page 135.
Model Container Utility	–	2022-B	2023-A	2023-B
ModelDesk	5.7	2022-B	2023-A	2023-B Refer to ModelDesk on page 139.
Model Interface Package for Simulink	22.1	2022-B	2023-A	2023-B Refer to Model Interface Package for Simulink on page 141.
MotionDesk	4.10	2022-B	2023-A	2023-B Refer to MotionDesk on page 143.
MotionDesk Blockset	2.6.5	2022-B	2023-A	2023-B Refer to MotionDesk on page 143.
Real-Time Testing	5.2	2022-B	2023-A	2023-B Refer to Real-Time Testing on page 145.
RTI ⁴⁾	7.18	2022-B	2023-A	2023-B Refer to RTI/RTI-MP and RTLib on page 147.
RTI-MP ⁵⁾	7.18	2022-B	2023-A	2023-B Refer to RTI/RTI-MP and RTLib on page 147.
RTI Bypass Blockset	3.18	2022-B	2023-A	2023-B Refer to RTI Bypass Blockset on page 149.
RTI CAN Blockset	3.4.14	2022-B	2023-A	2023-B

Product	dSPACE Release			
	2022-A	2022-B ¹⁾	2023-A	2023-B
RTI CAN MultiMessage Blockset	5.8	2022-B	2023-A	2023-B Refer to RTI CAN MultiMessage Blockset on page 151.
RTI Electric Motor Control Blockset	1.4.5	2022-B	2023-A	2023-B
RTI Ethernet Blockset	1.2.7	2022-B	2023-A	2023-B
RTI Ethernet (UDP) Blockset	1.4.7	2022-B	2023-A	2023-B
RTI LIN MultiMessage Blockset	3.8	2022-B	2023-A	2023-B Refer to RTI LIN MultiMessage Blockset on page 153.
RTI RapidPro Control Unit Blockset	2.2.7	2022-B	2023-A	2023-B
RTI Synchronized Time Base Manager Blockset	1.4.4	2022-B	2023-A	2023-B Refer to RTI Synchronized Time Base Manager Blockset on page 155.
RTI USB Flight Recorder Blockset	1.2.6	2022-B	2023-A	2023-B
RTI Watchdog Blockset	2.1.5	2022-B	2023-A	2023-B
SCALEXIO firmware	6.0	2022-B	2023-A	2023-B Refer to SCALEXIO Firmware on page 157.
Sensor Simulation	1.7	2022-B	–	–
Signal List Editor	–	–	–	2023-B Refer to Signal List Editor on page 161.
SYNECT	2.13	2022-B	2023-A	2023-B Refer to SYNECT on page 163.
SystemDesk	5.6	2022-B	2022-B	2023-B Refer to SystemDesk on page 167.
TargetLink	5.2	2022-B	2022-B	2023-B Refer to TargetLink on page 179.
VEOS	5.4	2022-B	2023-A	2023-B Refer to VEOS on page 251.

¹⁾ New versioning scheme of dSPACE products introduced.

²⁾ As of dSPACE Release 2021-B, ConfigurationDesk for RapidPro is no longer part of the dSPACE Release. You can download it from the dSPACE website as a separate software package with its own setup program. Refer to <https://www.dspace.com/go/CFDRP>.

³⁾ RTI FPGA Programming Blockset up to Release 2022-A.

⁴⁾ Including the standard I/O blocksets.

⁵⁾ Including the RTI Gigalink Blockset.

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 required migration steps.

New Key Product Features

Introduction

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

AutomationDesk

The new key features of AutomationDesk are:

- Support of Python 3.11
- Support of the new MicroLabBox II
- New Sleep block in the XIL API Convenience library

For more information on the new features, refer to [New Features of AutomationDesk 2023-B](#) on page 37.

Bus Manager (stand-alone)

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

- Transfer configured bus communication to bus configurations by copying, cutting, and pasting bus configuration elements
- Simulate PDU gateways (preview version)
- Generate missing ECUs for LIN PDUs
- Add event-controlled timing elements to IPDUs in communication matrices
- Enhanced J1939-22 support
- PDU Enable feature available for multiplexed IPDUs
- Counter Signal feature available for inspection

For more information, refer to [New Features of the Bus Manager \(Stand-Alone\) 2023-B](#) on page 57.

ConfigurationDesk

The new key features of ConfigurationDesk are:

- Support of the MicroLabBox II target platform, including adaption of the function block types to support the MicroLabBox II channel types.
- Support of the DS6242 D/A Board (SCALEXIO), including adaption of the function block types to support the DS6242 D/A Board channel types.
- Support of the new SCALEXIO FSX hardware for fault simulation.
- Portability of model containers between SCALEXIO and MicroLabBox II.
- Support of FMUs providing binary input/output variables.
- Support of FMUs providing terminals.
- Various enhancements of the Bus Manager for configuring bus communication for simulation, inspection, and manipulation purposes.

For more information, refer to [ConfigurationDesk](#) on page 65.

ControlDesk

The new key features of ControlDesk are:

General enhancements

- Support of the new MicroLabBox II
- Support of Python 3.11

Platform/device enhancements

- Platform support for the new MicroLabBox II
- SCALEXIO platform: Support of the DS6242 D/A Board and the DS6365-PE Automotive Ethernet Board
- GNSS device: New status variables, e.g., to indicate the number of satellites in use

For more information, refer to [New Features of Platform Management and Platforms/Devices \(ControlDesk 2023-B\)](#) on page 86.

Variable management enhancements

- Variable mapping improvements

For more information, refer to [New Variable Management Features \(ControlDesk 2023-B\)](#) on page 87.

Instrument enhancements

- Time Plotter, Index Plotter: Displaying signal names in the legend
- Time Plotter: Showing the connection name in time and data cursor tooltips
- Variable Array: Specifying custom header text
- Sound Controller: MP3 file support
- Hierarchy Array: Usability improvements
- Date and Time instrument category

For more information, refer to [New Instrument Features \(ControlDesk 2023-B\)](#) on page 88.

Bus Navigator enhancements

- Usage of port information from the AUTOSAR file for decoding SOME/IP-SD messages
- Importing and visualizing SOME/IP event parameters (preview feature)
- Customizing Bus Instrument generation (preview feature)
- J1939-22: Generating Compact and Manipulation Bus Instruments for Multi-PG IPDUs (Bus Manager applications only)
- J1939-22: Suspend PDU transmission manipulation support (Bus Manager applications only)
- PDU user port manipulation support (Bus Manager applications only)
- SecOC freshness overwrite value manipulation support (Bus Manager applications only)
- LIN schedule table access support (Bus Manager applications only)
- GTS communication manipulation support (dSPACE FlexRay Configuration Package-based applications only)
- Display of the PDU state in the region header (Bus Manager applications only)
- Visualizing signals for which 'ISignal Value' is enabled (Bus Manager applications only)

- Support of R22-11 ARXML files
- Usability improvements

For more information, refer to [New Bus Navigator Features \(ControlDesk 2023-B\)](#) on page 91.

Electrical error simulation enhancements

- Support of the new SCALEXIO FSX hardware for fault simulation

For more information, refer to [New Electrical Error Simulation Features \(ControlDesk 2023-B\)](#) on page 99.

User interface handling enhancements

- Searching for variables and block groups centrally
- Preview features

For more information, refer to [New User Interface Handling Features \(ControlDesk 2023-B\)](#) on page 99.

dSPACE ECU Flash Programming Tool

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

- Use CAN MAX_DLC value.

For more information on the new feature, refer to [New Features of the dSPACE ECU Flash Programming Tool 2023-B](#) on page 109.

dSPACE FlexRay Configuration Package

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

- Support of AUTOSAR Classic Platform R22-11.

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

- Enhancement to Create Corresponding ECU command.

For more information on the new feature, refer to [New Features of dSPACE FlexRay Configuration Package 2023-B](#) on page 111.

dSPACE XIL API

The new key features of dSPACE XIL API are:

- Support of Python 3.11
- MAPort: Support of the new MicroLabBox II
- EESPort: Support of the new SCALEXIO FSX hardware
- ECUPort: Captured data can now be stored in an MF4 file

For more information on the new features, refer to [New Features of dSPACE XIL API .NET 2023-B](#) on page 117.

ECU Interface Manager

The new key features of the ECU Interface Manager are:

- XCP time stamp synchronization
- Support of the new MicroLabBox II

For more information on the new features, refer to [New Features of ECU Interface Manager 2023-B](#) on page 121.

Firmware Manager

The new key features of the Firmware Manager are:

- Support of the new MicroLabBox II
- Changed behavior for the preselection of the SCALEXIO firmware archives

For more information, refer to [New Features of Firmware Manager 23.2](#) on page 125.

FPGA Programming Blockset

The new key features of the FPGA Programming Blockset 2023-B are:

- Extended FPGA design tool support.
- Support of MicroLabBox II.
- Enhancements to the FPGA frameworks.

For more information on the new features, refer to [New Features of the FPGA Programming Blockset 2023-B](#) on page 127.

Model Compare

The new key features of Model Compare 2023-B are:

- The new, filed-based accelerated model comparison method analyzes the pure model files directly and displays the comparison results immediately.
- The textconv driver with a built-in TargetLink support lets you analyze changes on TargetLink block properties and parameters in detail.

For more information on the new features, refer to [New Features of Model Compare 2023-B](#) on page 135.

ModelDesk

The new key feature of ModelDesk is:

- General: The core technology of ModelDesk is updated to prepare it for modern operating systems.
- Road Generator: You can import roads in the OpenDRIVE format 1.6, and the export quality is improved.

For more information on the new feature, refer to [New Features of ModelDesk 2023-B](#) on page 139.

Model Interface Package for Simulink

The new key features of Model Interface Package for Simulink are:

- Enhanced support of Simulink[®] In and Out blocks on the root-level of a behavior model.
- Support of MATLAB[®] R2023b

For more information on the new features, refer to [Model Interface Package for Simulink](#) on page 141.

Python Extensions

The new key feature of Python Extensions is:

- Support of Python 3.11

For more information, refer to [dSPACE Python Extensions](#) on page 115.

Real-Time Testing

The new key feature of Real-Time Testing is:

- Support of MicroLabBox II.

For more information on the new features, refer to [New Features of Real-Time Testing 2023-B](#) on page 145.

RTI, RTI-MP, and RTLib

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

- Support of MATLAB® R2023b.

For more information, refer to [New Features of RTI/RTI-MP and RTLib](#) on page 147.

RTI CAN MultiMessage Blockset

The new key feature of the RTI CAN MultiMessage Blockset is:

- Support of AUTOSAR Classic Platform R22-11.

For more information on the new features, refer to [New Features of the RTI CAN MultiMessage Blockset 2023-B](#) on page 151.

RTI LIN MultiMessage Blockset

The new key feature of the RTI LIN MultiMessage Blockset is:

- Support of AUTOSAR Classic Platform R22-11.

For more information on the new feature, refer to [New Features of the RTI LIN MultiMessage Blockset 2023-B](#) on page 153.

RTI Synchronized Time Base Manager Blockset

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

- Support of the new MicroLabBox II.

For more information on the new feature, refer to [New Features of the RTI Synchronized Time Base Manager Blockset 2023-B](#) on page 155.

SCALEXIO firmware

The new key feature of the SCALEXIO firmware is:

- Support of a new DS6242 D/A Board.
- Support of the DS6365-PE Ethernet Board.
- Support of the SCALEXIO FSX Hardware.

For more information on the new feature, refer to [New Features of the SCALEXIO Firmware 23.2](#) on page 157.

SystemDesk

The new key features of SystemDesk 2023-B are:

- A2L generation for AUTOSAR-compliant variable descriptions.
- XCP service integration according to the FMI Layered Standard.
- Support of the AUTOSAR R22-11 release.

For more information on the new features, refer to [New Features of SystemDesk 2023-B](#) on page 168.

TargetLink

The new key features of TargetLink 2023-B are:

- SOLE (Systems Of Linear Equations) Solver block to solve systems of linear equations
 - To provide a numerically efficient solution to problems like Kalman filters or least squares curve fitting.
- Dynamic element specification at Data Store Read and Data Store Write blocks
 - To dynamically access variables at Data Store Read and Data Store Write blocks using the Simulink® Enable indexing option for element selection and assignment.
- More C++ constructs
 - To use enumeration class data types, static casts, and namespaces on module level.
- Support of the Simulink® parameter Ensure output is virtual
 - To avoid differences in MIL simulation and in the generated code in simplified initialization mode, you can select the Ensure output is virtual checkbox at unenhanced Output blocks.
- Text-based DD files stored as .ddjson files
 - To offer the possibility to use DD files in version control systems and text-based workflows.

For more information on all new features, refer to [New Features of TargetLink 2023-B](#) on page 180.

For more information on the TargetLink migration aspects (TargetLink, TargetLink AUTOSAR Module, TargetLink Data Dictionary), refer to [Migrating to TargetLink 2023-B](#) and [TargetLink Data Dictionary 2023-B](#) on page 199.

VEOS

The new key features of VEOS are:

- Co-simulation with VEOS
- QEMU integration into VEOS simulation
- VEOS Simulator Console enhancements

For more information on the new features, refer to [New Features of VEOS 2023-B](#) on page 251.

New Use Scenario: Data Replay

Introduction

The tools of dSPACE Release 2023-B have new features to support the validation of ADAS/AD ECUs and AD software stack.

Validating sensor ECUs

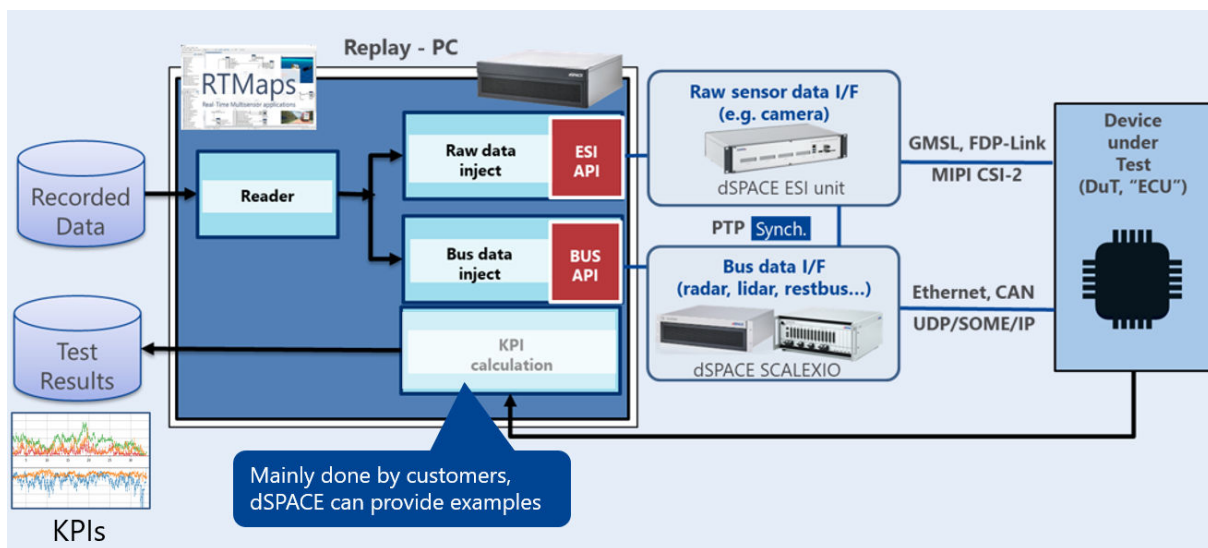
An ADAS/AD ECU can be a smart sensor, a domain controller or even a high performance computer. It is typically connected to a sensor front end, for example, camera, radar, or lidar or combination of all to perceive the physical environment and makes it available as digital raw data. The raw data is fed into

a perception and sensor fusion algorithm that detects the objects in the vicinity of the ego-vehicle. The detected objects can be processed within the same unit to generate an environment model of the vehicle or distributed further in the vehicle network so that a central processing unit can plan further actions.

You can validate real or virtual sensor ECUs using recorded data. The dSPACE tool chain is prepared for data replay in a hardware-in-the-loop (HIL) and software-in-the-loop (SIL) simulation. The ADAS/AD ECU can evaluate the data and you can compare the results with ground truth data by performing KPI (key performance indicator) calculation.

You can test a real ADAS/AD ECU in a hardware-in-the-loop (HIL) simulation. The ECU is connected to a HIL simulator (SCALEXIO system) that provides sensor information (e.g., lidar point cloud) via bus data. In addition, the ECU is connected to an Environment Sensor Interface (ESI) unit that provides the raw camera sensor data. The SCALEXIO system and ESI unit are connected to a replay PC that provides the data streams with time-stamped raw data and bus data. This can be done, for example, by RTMaps that reads the recorded real-world data and distributes it to the connected devices. The SCALEXIO system and ESI unit are exactly synchronized to the SCALEXIO system time using the precision time protocol (PTP) over Ethernet.

You can test the software stack of an ADAS/AD ECU (virtual sensor ECU) in a software-in-the-loop (SIL) simulation. Such tests can be performed on a PC or in a cloud environment without any dedicated hardware. The software of the sensor ECU must be embedded in a V-ECU. The bus communication must be embedded in a bus simulation container (BSC). RTMaps reads the recorded real-world data and distributes it to the BSC.



Related topics

Basics

[Replaying Data to Validate Real Sensor ECUs \(SCALEXIO – Hardware and Software Overview\)](#)

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 2023-B

Introduction

After you install Release 2023-B, some additional steps might be required.

Migrating from dSPACE Release 2023-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 2022-B or earlier

To migrate from dSPACE Release 2022-B or earlier to Release 2023-B, you also have to perform the migration steps of the intervening dSPACE Releases. All of the required migration steps can be performed using software from dSPACE Release 2023-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 represents 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 2023-B\Print\PreviousReleases`.
- On the dSPACE DVDs. Refer to `\Doc\PreviousReleases`.

- At <https://www.dspace.com/go/migration> for download. Here, you can also find *New Features and Migration* documents for very early Releases.

Migrating Python Scripts from Python 3.9 to Python 3.11

Where to go from here

Information in this section

Main Changes in Python 3.11	29
General information on Python 3.11.	
Main Changes in Handling Python 3.11 with dSPACE Software	30
Overview of the packages in the Python 3.11 distribution that is contained in the dSPACE Release 2023-B.	
General Information on Using Python Installations	31
Information on using Python installations in parallel.	
Technical Changes	33
The migration to Python 3.11 has caused changes in the Python API.	
Product-Specific Information on Migrating	33

Main Changes in Python 3.11

Reasons for using Python 3.11 instead of Python 3.9

The final release of Python 3.11 has been available since October 2022 and the end of life of Python 3.9 is scheduled for October 2025. The latest Python release with its included Python packages provides new features, security updates, and bugfixes.

Therefore, dSPACE switches to Python 3.11 with dSPACE Release 2023-B.

Related documents available on the Python website

For a short overview of the main changes in Python 3.11, refer to [Technical Changes](#) on page 33. For information on all the changes in the Python language and environment from Python 3.9 to Python 3.11, refer to www.python.org.

The Python Software Foundation provides the following documents:

- What's New from Python 3.9 to 3.10:
<https://docs.python.org/3.10/whatsnew/3.10.html>
- What's New from Python 3.10 to 3.11:
<https://docs.python.org/3.11/whatsnew/3.11.html>

Related topics

Basics

Technical Changes	33
---	----

Main Changes in Handling Python 3.11 with dSPACE Software

Introduction

With Python 3.11, some aspects of handling Python with dSPACE software have changed.

Libraries

The packages used with Python 3.11 and distributed on dSPACE DVDs have changed as shown in the following table.

Package	Python 3.9 (Release 2023-A)	Python 3.11 (Release 2023-B)
certifi	2022.12.7	2023.7.22
cffi	-	1.15.1 (new)
charset-normalizer	3.0.1	3.2.0
clr-loader	-	0.2.5 (new)
comtypes	1.1.14	1.2.0
countourpy	1.0.6	1.1.0
Core	3.9.16.1 ¹⁾	3.11.5
cycler	0.11.0	0.11.0
fonttools	4.38.0	4.42.1
future	0.18.3	0.18.3
grpcio	1.51.1	1.57.0
grpcio_tools	1.51.1	1.57.0
idna	3.4	3.4
kiwisolver	1.4.4	1.4.5
lxml	4.9.2	4.9.3
matplotlib	3.6.3	3.7.2
numpy	1.24.1	1.25.2
packaging	23.0	23.1
pillow	9.4.0	10.0.0

Package	Python 3.9 (Release 2023-A)	Python 3.11 (Release 2023-B)
pip	22.3.1	23.2.1
protobuf	4.21.12	4.24.2
pyparser	2.21	2.21
pyglet	2.0.3	2.0.9
yparsing	3.0.9	3.0.9
pypubsub	4.0.3	4.0.3
Python-dateutil	2.8.2	2.8.2
pythonnet	2.5.3 ¹⁾	3.0.10 ¹⁾
pytz	2020.7	2023.3
pywin32	305.10 ¹⁾	306.10 ¹⁾
requests	2.28.2	2.31.0
scipy	1.10.0	1.11.2
six	1.16.0	1.16.0
urllib3	1.26.14	2.0.4
wxPython	4.2.0	4.2.1
yapsy	1.12.2	1.12.2

¹⁾ This component contains some dSPACE-specific bugfixes. It should not be replaced by the component from the standard Python distribution.

General Information on Using Python Installations

Introduction

If you work with dSPACE software from dSPACE Release 2023-A or earlier, which supports Python 3.9, and dSPACE software from dSPACE Release 2023-B, which supports Python 3.11, both Python versions are installed on the PC and can be used in parallel.

Limitations when using Python 3.9 and Python 3.11 in parallel

You can use both Python versions in parallel. However, you must observe the following limitations:

- You can set the file associations for PY and PYW files to only one Python version. This is usually the latest Python version you installed.

Note

If you install dSPACE Release 2023-A or earlier after the installation of dSPACE Release 2023-B, Python 3.9 will be installed last. Therefore, PythonWin 3.9 is registered to open Python files. To set the file associations to PythonWin 3.11, execute the **PythonInstaller.exe** from <DVDRoot>\Products\Common\Python3.11 again.

To open a Python file with PythonWin 3.11 temporarily, use the **Edit with PythonWin 3.11** command on the context menu of the file.

- Environment variables are used by both Python versions. You must set their values, for example, for PYTHONHOME, to the Python installation you want to work with. For an overview of environment variables set by Python, refer to: <https://docs.python.org/3.11/using/windows.html>.

Note

To avoid unintentional effects, which can be difficult to identify, you are recommended to not use environment variables. For the same reason, you should not extend the system path by a Python installation path.

Using dSPACE test automation with both Python versions in parallel

If a test automation script uses dSPACE Python Modules and you do not want to migrate the script, you have to work with both Python versions in parallel. The dSPACE Python Modules for Python 3.9 are available up to and including Release 2023-A.

Note

It is recommended to use only dSPACE software from the same Release when using Python scripts. Using both Python versions in the same application context, such as automating the access to an older version of ControlDesk via AutomationDesk or using AutomationDesk with an activated Real-Time Testing version that does not support Python 3.11, might cause a conflict.

Technical Changes

Introduction

Migrating from Python 3.9 to Python 3.11 has caused changes in the Python API.

Note

Independently of the number of required manual modifications, you must test the migrated code.

Migration issues

The following changes in pythonnet 3.0 from Python 3.11 require manual migration of your Python scripts.

pythonnet: Change of the return values of .NET methods If you use pythonnet 3.0 in Python 3.11, .NET methods that return an interface are automatically wrapped in that interface. Therefore, you cannot access members that are only part of the implementation class. You can only access members that are part of the interface.

For information on the product-specific migration steps, refer to [Product-Specific Information on Migrating](#) on page 33.

pythonnet: Implicit type conversion of enumeration values is disabled The implicit conversion from C# enums to Python int and vice versa has been disabled. If the Enum object does not provide a member with the related integer value, you must either use an enum member, e.g., `MyEnum.Option`, or an enum constructor, e.g., `MyEnum(<intValue>)`.

pythonnet: .NET collections and arrays are no longer automatically converted to Python collections The automatic conversion of .NET collections and arrays introduced with pythonnet 2.4 is undone with pythonnet 3.0. You must therefore migrate the methods and properties of a Python collection to the methods and properties of the .NET collection. For example, you must use `MyObject.Count` instead of `len(MyObject)`.

Product-Specific Information on Migrating

Products with editable Python content

Python scripts are used explicitly or implicitly in the following dSPACE products:

- AutomationDesk
- Bus Manager (stand-alone)
- ConfigurationDesk
- ControlDesk
- ModelDesk
- dSPACE Platform API Package (Python Extensions, XIL API .NET, Platform Management API)
- Real-Time Testing

- SYNECT
- SystemDesk

No other dSPACE products are affected by the Python migration.

You must check all custom scripts of the above products for required migration, e.g., custom user functions. For further product-specific information, see below.

Using AutomationDesk

Due to the disabled Implicit type conversion of enumeration values in pythonnet 3.0, the handling of XIL API Enums has changed.

To use XIL API Enums in the Python code of Exec blocks, you must use the specified Enum object instead of the related integer value.

Example: The following example shows how to specify seconds as the duration unit for a capture with pythonnet 2.5.3 in Python 3.9 and the migrated code for pythonnet 3.0 in Python 3.11.

```
# pythonnet 2.5.3
Capture.DurationUnit = 0
# pythonnet 3.0
Capture.DurationUnit = xilapi.Common.Capturing.Enum.DurationUnit.eSECOND
```

Using Bus Manager (stand-alone)

There are no product-specific migration issues to consider.

Using ConfigurationDesk

There are no product-specific migration issues to consider.

Using ControlDesk

There are no product-specific migration issues to consider.

Using dSPACE Platform API Package

Using Python Extensions There are no product-specific migration issues to consider.

Using Platform Management API There are no product-specific migration issues to consider.

Using XIL API .NET The changed behavior of the .NET methods returning an interface object is internally adapted in XIL API .NET by using codecs. You do not have to migrate your scripts.

Example: The read method of the MAPort is defined as

`IBaseValue Read(variableName)`. If you use pythonnet 3.0, executing `curValue = MAPort.Read(variableName)` returns `curValue` as an `IBaseValue`.

- Reading `curValue`, e.g., by executing `val = curValue.Value`, causes an exception, because the `IBaseValue` interface does not provide the `Value` property.
- Writing `curValue`, e.g., by executing `curValue.Value = val`, does not modify the object, but Python creates a new property *without* throwing an

exception. It is therefore difficult to detect the incorrect behavior of your application due to the changed implementation in pythonnet 3.0.

To avoid this potential hidden incorrect behavior of your application, the following interfaces in the dSPACE XIL API .NET implementation are internally migrated to their implementation class:

- IBaseError
- IBaseErrorBuilder
- IBaseValue
- ICapture
- ICaptureEvent
- ICompuMethod
- IDataFileSegment
- IDuration
- IEESPort
- IEESPortConfig
- IErrorSet
- IMAPort
- IMAPortConfig
- IMatrixValue
- IPort
- IScalarValues
- ISignalDescription
- ISignalSegment
- ISymbol
- ITestbench
- IVectorValue
- IWatcher

Using ModelDesk

There are no product-specific migration issues to be concerned.

Using Real-Time Testing

In Real-Time Testing, Python is used on the host PC and on the real-time platform.

The interpreters support different Python versions:

- Python 3.11:
 - Interpreter on the host PC
- Python 3.6:
 - Interpreter used with VEOS as of dSPACE Release 2019-B
 - Interpreter on MicroAutoBox III as of dSPACE Release 2019-B
 - Interpreter on SCALEXIO/DS6001 as of dSPACE Release 2020-B
 - Interpreter on MicroLabBox II as of dSPACE Release 2023-B
- Python 2.7.11:
 - All other real-time platforms

You must migrate your custom scripts to Python 3.11 manually. Real-Time Testing does not provide any features for the automatic migration of custom scripts from Python 3.9 to Python 3.11.

The activated Real-Time Testing version and the dSPACE software using Real-Time Testing, e.g., AutomationDesk, must support the same Python version.

Using SYNECT

The new pythonnet 3.0 version distributed with Python 3.11 can cause script migration issues due to stricter script handling, e.g., allowing for less implicit type conversions.

Using the `SNDataExchange` .NET DLL for custom Python adapters is a common use case of pythonnet with SYNECT. Refer to [Basics on Implementing Adapter Classes \(SYNECT Guide !\[\]\(23d9fc146e83b5c3013cfa32c784f8d5_img.jpg\)](#))

Note

SYNECT server scripts are not affected because SYNECT does not use Python 3.11 but an IronPython interpreter for server scripts.

Using SystemDesk

The SystemDesk demo scripts did not need to be modified for Python 3.11. It is therefore expected that custom scripts do not need migration either.

AutomationDesk

Where to go from here

Information in this section

New Features of AutomationDesk 2023-B.....	37
Migrating to AutomationDesk 2023-B.....	38

New Features of AutomationDesk 2023-B

General enhancements

Support of Python 3.11 AutomationDesk supports Python 3.11 and the included pythonnet 3.0 package. For information on the required migration steps, refer to [Migrating Python Scripts from Python 3.9 to Python 3.11](#) on page 29.

Support of new dSPACE hardware The AutomationDesk Platform Manager lets you register the new MicroLabBox II and download simulation applications to the platform. For more information on the new MicroLabBox II, refer to [General Enhancements and Changes](#) on page 13.

Enhancements to the XIL API Convenience library

Sleep block using the simulation platform time The XIL API Convenience library now provides a Sleep block that uses the time information from the simulation platform on which the application is executed. The simulation platform is represented by the referenced MAPort data object. In offline mode, or if the MAPort data object is not initialized, the behavior of the Sleep block is equivalent to the Sleep block in the Main Library, which uses the system clock of the host PC. For more information, refer to [Sleep \(XIL API\) \(AutomationDesk Accessing Simulation Platforms !\[\]\(6bb0e4f14c4133b37d2887cb37e67ddd_img.jpg\)\)](#).

Migrating to AutomationDesk 2023-B

General migration aspects

If you open an AutomationDesk project with a later AutomationDesk version, the software automatically detects whether migration is required. Click OK in the message dialog to start the migration. Save the migrated project to another path or name.

Note

Before you open an earlier 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 !\[\]\(10f8862fc183b400327470ea85afe9ae_img.jpg\)](#)).

For more information, refer to [Migration \(AutomationDesk New Features and Migration !\[\]\(e1d6102fe77919492c04879c8450f1f5_img.jpg\)](#)).

Note

With dSPACE Release 2023-B, AutomationDesk supports Python 3.11. For information on changes and migration aspects of Python scripts in AutomationDesk, refer to [Product-Specific Information on Migrating](#) on page 33.

Discontinued support of ASAM XIL API 2.1

AutomationDesk 2023-B no longer supports XIL API servers, which are based on ASAM XIL API 2.1 and earlier. This relates to servers based on dSPACE XIL API .NET and third-party servers.

You can use dSPACE XIL API servers as of dSPACE Release 2022-A based on ASAM XIL API 2.2.

Automotive Simulation Models (ASM)

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

All ASM Products.....	40
ASM Battery.....	41
ASM Diesel Engine.....	43
ASM Drivetrain Basic.....	44
ASM Electric Components.....	45
ASM Fuel Cell.....	46
ASM Gasoline Engine.....	48
ASM Traffic.....	49
ASM Trailer.....	50
ASM Truck.....	52
ASM Vehicle Dynamics.....	54

All ASM Products

Where to go from here

Information in this section

Changes in All ASM Demo Models.....	40
Migrating All ASM Blocksets.....	40

Changes in All ASM Demo Models

Real-time operating system support for SCALEXIO

From now on, the SCALEXIO processing hardware only provides a 64-bit Linux-based real-time operating system. Therefore, the ASM demo models no longer support a 32-bit, but only the 64-bit real-time operating system.

Communication signals

The structure of the ASM modules has been improved. The interfaces used for communication with other modules have been moved to the top level of every ASM module. This lets you adapt the model more easily. Also, for this purpose, the structure in the Interface_Out has been optimized.

Modification of initialization files

The initialization files of the ASM demo models have been modified.

The following information fields have been removed:

- 'Author'
- 'Version'
- 'LastModified'

The initialization files of migrated models are not changed.

Migrating All ASM Blocksets

Operator library blocks in demo models

All operator library blocks in a demo model are linked to the open standard ASM libraries. The OPERATOR and DEVELOPER buttons in the ASM Model Tools for switching between both library versions are removed.

ASM Battery

Where to go from here

Information in this section

New Features of ASM Battery Blockset 2023-B.....	41
Changes in the ASM Battery Demo Models.....	41
Migrating to ASM Battery Blockset 2023-B.....	42

New Features of ASM Battery Blockset 2023-B

CHARGING_TIME block

This is a new block. It calculates the remaining battery charging time dependent on the charging power and battery capacity.

BATTERY_PACK_DELTA_VOLTAGE block

The ASMSignalBus has been extended with battery capacity in kWh.

The power loss signal vector has been changed to a one-dimensional signal in the ASMSignalBus.

Now it is possible to differentiate between *reset* and *SOC reset*. With SOC reset, only the state of charge is reset.

BATTERY_PACK_VOLTAGE block

The ASMSignalBus has been extended with battery capacity in kWh.

Now it is possible to differentiate between *reset* and *SOC reset*. With SOC reset, only the state of charge is reset.

CSC_INTERFACE_IN block

This is a new block. It interfaces signals coming from the plant model, for example, a battery, to the cell supervision circuit (CSC).

CSC_INTERFACE_OUT block

This is a new block. It interfaces signals going from the cell supervision circuit (CSC) to the plant model, for example, a battery.

Changes in the ASM Battery Demo Models

Charging power relays

The vehicle charging control ECU now simulates two power relays to control the cable connection on the charging station as well the high voltage battery side.

Charging time	The battery management system has been extended with a calculation of the battery charging time.
Battery management system soft ECU	<p>The battery management system (BMS) has been restructured to represent a distributed topology. It now consists of a central battery management unit (BMU), which mainly controls the DC-link and estimates the state-of-charge and several cell supervision circuit (CSC) modules.</p> <p>The main simulated tasks of a CSC are cell monitoring and balancing. The number of supervised cells per CSC is determined automatically by dividing the number of active cells per number of CSC modules.</p> <p>The ASM Battery demo model comes with six CSC modules. You can easily change this number to fit your requirements.</p>

Migrating to ASM Battery Blockset 2023-B

BATTERY_PACK_VOLTAGE block	The new <i>SOC reset</i> input is connected with a zero-constant.
BATTERY_PACK_DELTA_VOLTAGE block	The new <i>SOC reset</i> input is connected with a zero-constant.

Related topics

Basics

[Migrating ASM Models \(ASM User Guide !\[\]\(a870788d6ed9b8fd294b7654a8c8526b_img.jpg\)\)](#)

ASM Diesel Engine

Migrating to ASM Diesel Engine Blockset 2023-B

MASS_FRACTION block

The mass fraction of unburned air is set to 1 if all engine mass flows are 0 . The former behavior of setting it to 0 is restored during migration.

Related topics**Basics**

[Migrating ASM Models \(ASM User Guide !\[\]\(e474458956c9a37fbf9586ddb60a7fa1_img.jpg\)\)](#)

ASM Drivetrain Basic

Where to go from here

Information in this section

New Features of ASM Drivetrain Basic Blockset 2023-B.....	44
Migrating to ASM Drivetrain Basic Blockset 2023-B.....	44

New Features of ASM Drivetrain Basic Blockset 2023-B

TORQUE_INTERVENTION_ **CONTROL block**

The issue that the first gear request during the simulation is delayed by one step size compared to the successive gear requests has been fixed.

Migrating to ASM Drivetrain Basic Blockset 2023-B

TORQUE_INTERVENTION_ **CONTROL block**

The block is migrated so that the former behavior remains unchanged. To use the new block feature, remove the migration block and connect the respective ports.

Related topics

Basics

[Migrating ASM Models \(ASM User Guide !\[\]\(626ce8ac21792b9405bfddfea8e0c96a_img.jpg\)\)](#)

ASM Electric Components

Where to go from here

Information in this section

New Features of ASM Electric Components Blockset 2023-B.....	45
Changes in the ASM Electric Components Demo Models.....	45
Migrating to ASM Electric Components Blockset 2023-B.....	45

New Features of ASM Electric Components Blockset 2023-B

ELECTRIC_MACHINE_BASIC block

Different torque speed curves are now used to parameterize the model in *generator* and in *motor* mode separately. Also, the machine efficiency is now calculated based on the efficiency tables or calculated using only the copper losses.

Changes in the ASM Electric Components Demo Models

Battery Electric and Hybrid Vehicle demo models

The vehicle charging control ECU now simulates two power relays to control the cable connection on the charging station as well the high voltage battery side.

AURELION Manager project

The demo models qualified for animation now include an AURELION Manager project. This allows you to easily start an AURELION animation of the demo models.

Migrating to ASM Electric Components Blockset 2023-B

ELECTRIC_MACHINE_BASIC block

The model is extended with new parameters.

Related topics

Basics

[Migrating ASM Models \(ASM User Guide !\[\]\(4436e6b00b9d5e62c2a161129eb3e4d0_img.jpg\)\)](#)

ASM Fuel Cell

Where to go from here

Information in this section

New Features of ASM Fuel Cell Blockset 2023-B.....	46
Changes in the ASM Fuel Cell Demo Model.....	47
Migrating to ASM Fuel Cell Blockset 2023-B.....	47

New Features of ASM Fuel Cell Blockset 2023-B

ELECTROCHEMISTRY_ **PHYSICAL block**

The block now lets you configure a short circuit resistance to model internal currents. In addition, the processing function for calculating the block parameters from standardized measurement data has been optimized and extended to include the new membrane short circuit resistance. As state-of-the-art fuel cell shutdown routines use low cell voltage as the threshold, the block now allows for the simulation of the polarization curve for cell voltages below 0.5V with a semi-empirical modeling approach.

GAS_DIFFUSION_LAYER block

The block now supports hydrogen crossover, which is the diffusion of hydrogen across the membrane. It is assumed that the entire hydrogen crossover completely reacts with oxygen to water on the cathodic catalyst layer if oxygen is available. Otherwise, hydrogen diffuses further to cathodic gas flow channels.

MAPPING_MDOT_HDOT block

The block now supports hydrogen crossover, which is the diffusion of hydrogen across the membrane. It is assumed that the entire hydrogen crossover completely reacts with oxygen to water on the cathodic catalyst layer if oxygen is available. Otherwise, hydrogen diffuses further to cathodic gas flow channels.

MEMBRANE block

The block now supports hydrogen crossover, which is the diffusion of hydrogen across the membrane. It is assumed that the entire hydrogen crossover completely reacts with oxygen to water on the cathodic catalyst layer if oxygen is available. Otherwise, hydrogen diffuses further to cathodic gas flow channels.

STACK_ASSEMBLY block

The block functionality has been extended to model heat exchange in the stack due to conduction and heat exchange with the environment due to radiation and free convection.

Changes in the ASM Fuel Cell Demo Model

Charging power relays	The vehicle charging control ECU now simulates two power relays for controlling the cable connection on the charging station as well the high-voltage battery side.
Automatic stack layout generation	<p>The ASM Fuel Cell demo model now comes with a Python script that creates the stack layout for different cell configurations. The Python script is part of the ControlDesk project of the demo model.</p> <p>When the script is executed, the required information of the actual configuration is read from the real-time or VEOS application running on the platform at that moment.</p>

Migrating to ASM Fuel Cell Blockset 2023-B

ELECTROCHEMISTRY_PHYSICAL block	The library link of the block is changed to ELECTROCHEMISTRY_PHYSICAL_2_0.
GAS_DIFFUSION_LAYER block	<p>Four Constant blocks with the value <i>zero</i> are connected to the new inports and three Terminator blocks are connected to the new outports.</p> <p>A time constant parameter is added to the block for the first-order dynamics of the substance flux through the gas diffusion layer.</p>
MEMBRANE block	<p>Six Constant blocks with the value <i>zero</i> are connected to the new inports and eight Terminator blocks are connected to the new outports.</p> <p>The diffusion coefficient of hydrogen in the membrane is added to the block as a new parameter.</p>
MAPPING_MDOT_HDOT block	Five Constant blocks with the value <i>zero</i> are connected to the new inports.

Related topics

Basics

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

ASM Gasoline Engine

Migrating to ASM Gasoline Engine Blockset 2023-B

MASS_FRACTION block

The mass fraction of unburned air is set to 1 if all engine mass flows are 0 . The former behavior of setting it to 0 is restored during migration.

Related topics

Basics

[Migrating ASM Models \(ASM User Guide !\[\]\(17acf1afa8cdf0b67c53d4865a5ed469_img.jpg\)\)](#)

ASM Traffic

Changes in the ASM Traffic Demo Model

AURELION Manager project

The demo model now includes an AURELION Manager project. This allows you to easily start an AURELION animation of the demo model.

ASM Trailer

Where to go from here

Information in this section

New Features of ASM Trailer Blockset 2023-B.....	50
Changes in the ASM Trailer Demo Model.....	51

New Features of ASM Trailer Blockset 2023-B

**SUSPENSION_FORCE_
KINEMATICS_TRAILER_
FRONT** block

The last block output has been renamed to `gF_dq_Front_dt[2x2][]`

**SUSPENSION_FORCE_
KINEMATICS_TRAILER_
REAR** block

The last block output has been renamed to `gF_dq_Rear_dt[2x2][]`.

**SUSPENSION_FORCE_
KINEMATICS_TRAILER_
REAR_2ND** block

The last block output has been renamed to `gF_dq_Rear_dt[2x2][]`.

**SUSPENSION_FORCE_
KINEMATICS_TRAILER_
REAR_3RD** block

The last block output has been renamed to `gF_dq_Rear_dt[2x2][]`.

**TIRE_MODEL_MAGIC_
FORMULA** block

The unloaded tire radius and constant vertical tire stiffness have been added to the ASMSignalBus of all tire models.

TIRE_MODEL_TMEASY block

The unloaded tire radius and constant vertical tire stiffness have been added to the ASMSignalBus of all tire models.

Changes in the ASM Trailer Demo Model

AURELION Manager project

The demo model now includes an AURELION Manager project. This allows you to easily start an AURELION animation of the demo model.

ASM Truck

Where to go from here

Information in this section

New Features of ASM Truck Blockset 2023-B.....	52
Changes in the ASM Truck Demo Model.....	52

New Features of ASM Truck Blockset 2023-B

SUSPENSION_FORCE_ KINEMATICS_REAR_2ND block

The last block output has been renamed to gF_dq_Rear_dt[2x2][].

SUSPENSION_FORCE_ KINEMATICS_REAR_3RD block

The last block output has been renamed to gF_dq_Rear_dt[2x2][].

TIRE_MODEL_MAGIC_ FORMULA block

The unloaded tire radius and constant vertical tire stiffness have been added to the ASMSignalBus of all tire models.

TIRE_MODEL_TMEASY block

The unloaded tire radius and constant vertical tire stiffness have been added to the ASMSignalBus of all tire models.

Changes in the ASM Truck Demo Model

AURELION Manager project

The demo models now includes an AURELION Manager project. This allows you to easily start an AURELION animation of the demo model.

New Truck Trailer with Traffic demo model

The ASM Truck library now provides a Truck Trailer with Traffic demo model. The ASM traffic components extend the Truck Trailer demo with the opportunity to simulate the movement of fellow vehicles as well as several sensor models equipped to the ASM vehicle.

Similar to the other ASM demo models, the new demo model includes projects for:

- ASM
- ModelDesk

- ControlDesk
- AURELION Manager

ASM Vehicle Dynamics

Where to go from here

Information in this section

New Features of ASM Vehicle Dynamics Blockset 2023-B.....	54
Changes in the ASM Vehicle Dynamics Demo Model.....	54
Migrating to ASM Vehicle Dynamics Blockset 2023-B.....	55

New Features of ASM Vehicle Dynamics Blockset 2023-B

ADAMS2ASM converter

The ADAMS2ASM converter now provides a new user interface for automated parameterization of a vehicle using ADAMS simulation. It now lets you convert data from Adams Car® and scenarios to ModelDesk automatically and validate the simulation results.

SUSPENSION_FORCE_ KINEMATICS_FRONT block

The last output was renamed to gF_dq_Front_dt[2x2][].

SUSPENSION_FORCE_ KINEMATICS_REAR block

The last output was renamed to gF_dq_Rear_dt[2x2][].

TORQUE_INTERVENTION_ CONTROL block

The issue where the first gear request during the simulation is delayed by one step size compared to the successive gear requests has been fixed.

TIRE_MODEL_MAGIC_ FORMULA block

The unloaded tire radius and constant vertical tire stiffness have been added to the ASMSignalBus of all tire models.

TIRE_MODEL_TMEASY block

The unloaded tire radius and constant vertical tire stiffness have been added to the ASMSignalBus of all tire models.

Changes in the ASM Vehicle Dynamics Demo Model

AURELION Manager project

The demo model now includes an AURELION Manager project. This allows you to easily start an AURELION animation of the demo model.

Migrating to ASM Vehicle Dynamics Blockset 2023-B

TORQUE_INTERVENTION_ **CONTROL block**

The block is migrated so that the former behavior remains unchanged. To use the new block feature, remove the migration block and connect the respective ports.

Related topics

Basics

[Migrating ASM Models \(ASM User Guide !\[\]\(8d0f0e0fe25b320c33272c52aec1fbca_img.jpg\)\)](#)

Bus Manager (Stand-Alone)

Where to go from here

Information in this section

New Features of the Bus Manager (Stand-Alone) 2023-B.....	57
Migrating to the Bus Manager (Stand-Alone) 2023-B.....	60
Migrating Projects and Applications Created with Previous Versions.....	60
Bus Manager (Stand-Alone) Discontinuations.....	63

New Features of the Bus Manager (Stand-Alone) 2023-B

New supported AUTOSAR Release

The Bus Manager (stand-alone) now supports AUTOSAR system description files based on AUTOSAR Classic Platform Release R22-11.

Transferring configured bus communication to bus configurations

The Bus Manager (stand-alone) now lets you easily transfer configured bus communication from one bus configuration to another. To do this, you can copy entire bus configurations, or copy or cut individual bus configuration elements and paste them to one or more bus configurations in the Bus Configurations table.

For more information, refer to [Copying, Cutting, and Pasting Bus Configuration Elements \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(cf531ed27e91483460120fcc057b3901_img.jpg\)](#)).

Simulating PDU gateways (preview version)

The Bus Manager (stand-alone) now lets you simulate PDU gateways. With PDU gateways, an ECU can route the data of PDUs from one or more communication clusters to other communication clusters. PDU gateways must be specified in communication matrices. If required, you can specify user-defined PDU gateways by modifying communication matrices. However, to simulate PDU gateways, you have to enable the PDU gateway support for bus configurations.

For more information, refer to [Simulating PDU Gateways \(Preview Version\) \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(529949c2c3dadbaa4e538e8c643454bc_img.jpg\)](#)).

Note

The current version of simulating PDU gateways is a preview version. For more information on preview versions of Bus Manager features, refer to [Overview of the Bus Manager \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(99f58673407353e96a019fbca558fd72_img.jpg\)](#)).

Generating missing ECUs for LIN PDUs

The Bus Manager (stand-alone) now generates missing ECUs for LIN PDUs. The ECUs are generated if the communication matrix does not specify the sending or receiving ECU or if a specified ECU is not part of the communication matrix.

For more information, refer to [Working with Communication Matrices \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(a870788d6ed9b8fd294b7654a8c8526b_img.jpg\)](#)).

Adding event-controlled timing elements to IPDUs in communication matrices

The Bus Manager (stand-alone) now lets you add missing event-controlled timing elements to ISignal IPDUs and extended multiplexed IPDUs in communication matrices. By this, you can specify an event-controlled transmission if it is not specified yet in the communication matrix.

For more information, refer to [Adding Event-Controlled Timing Elements to IPDUs \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(6059a5aa8b4ca7bb793408023d6c6e42_img.jpg\)](#)).

Enhanced J1939-22 support

The Bus Manager (stand-alone) now provides the following enhancements for J1939-22:

Manipulation of J1939-22-compliant IPDUs The Bus Manager (stand-alone) now lets you manipulate J1939-22-compliant IPDUs with a payload length of up to 60 bytes, i.e., IPDUs that are transmitted using the Multi-PG protocol. You can assign such IPDUs and their ISignals to the Manipulation part of a bus configuration and add the PDU User Code, ISignal Overwrite Value, and ISignal Offset Value manipulation features to them.

Moreover, the Bus Manager (stand-alone) provides the new Suspend PDU Transmission manipulation feature. The feature lets you suspend the transmission of J1939-22-compliant IPDUs that are transmitted using the Multi-PG protocol.

- For an overview of the features, refer to [Basics on Bus Configuration Features \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(098e47036f78288d477e334896a43770_img.jpg\)](#)).
- For more information on the Suspend PDU Transmission feature, refer to [Suspending the Transmission of PDUs \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(4eae4ca9e3bdd839cea72df3843a6754_img.jpg\)](#)).

Support of assurance data The Bus Manager (stand-alone) now supports assurance data according to the Multi-PG and Broadcast Announce Message (BAM) protocols for J1939-22-compliant IPDUs, i.e., the Bus Manager (stand-alone) can transmit and receive assurance data.

For more information, refer to [Aspects of the J1939 Protocol \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(1d3a1175dd4902218e694b9c098adb83_img.jpg\)](#)).

Configurable communication matrix settings for J1939-22-compliant IPDUs The Bus Manager (stand-alone) now lets you modify the type of service, trailer format, assurance data type, and assurance data size for J1939-22-compliant IPDUs.

For more information, refer to [Configurable Settings of PDUs \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(4729e517bc6a7cd81c8025b9646574fb_img.jpg\)](#)).

PDU Enable feature available for multiplexed IPDUs

The Bus Manager (stand-alone) now lets you add the PDU Enable feature to multiplexed IPDUs.

For more information, refer to [Enabling and Disabling the Transmission of PDUs \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(cbe2492b119e39e02a1dab2af4a4b296_img.jpg\)](#)).

Counter Signal feature available for inspection

The Bus Manager (stand-alone) now provides the Counter Signal feature for ISignals that are assigned to the Inspection part of a bus configuration.

For more information, refer to [Working with Counter Signals \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(5361750c22c4e047a52f4eac1ec2d4cc_img.jpg\)](#)).

Unused bit pattern of container IPDUs with a dynamic layout for padding bytes

The Bus Manager (stand-alone) now uses the unused bit pattern of container IPDUs with a dynamic layout as the value for padding bytes of the related frames. If required, padding bytes are added to frames to extend the frame length to a valid CAN FD length.

For more information, refer to [Aspects of Miscellaneous Supported CAN Bus Features \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(0d5ec72f61334709c3fc9450209b754f_img.jpg\)](#)).

Enhancements of the Restbus Configuration Exclude List

The Bus Manager (stand-alone) now lets you add ECUs to the Restbus Configuration Exclude List via drag & drop. When you do this, the ECUs are semicolon-separated in the list and are excluded from the restbus configuration when you create it.

For more information, refer to [Creating Restbus Configurations \(Bus Manager \(Stand-Alone\) Implementation Guide !\[\]\(2bae76de5ebbd5c4d7d47162f1673734_img.jpg\)](#)).

Redesigned Options dialog

The Bus Manager (stand-alone) now provides a fully redesigned Options dialog. The Options dialog lets you find options quickly by a category-based navigation and a search bar with direct response while typing. In addition, the Options dialog now contains the new Search Path Editor, which allows a graphical configuration of search paths for artifacts used by the Bus Manager (stand-alone).

For more information, refer to [Options \(ConfigurationDesk User Interface Reference !\[\]\(28f72b996fc97883dfd9d4e8b1b16b4e_img.jpg\)](#)).

New features of the tool automation interface

The ConfigurationDesk automation interface supports additional features. For more information, refer to [New Features and Changes to the Automation Interface for Release 2023-B \(ConfigurationDesk Automating Tool Handling !\[\]\(2bdfe261b986065ee0ac76460d6528c9_img.jpg\)](#)).

Migrating to the Bus Manager (Stand-Alone) 2023-B

Changed order in TRC files and A2L files

Groups, blocks, and variables in TRC files and A2L files are now sorted alphabetically. As a consequence, you might have to adapt your custom tools based on these files.

Migrating to new ID behavior of Simulink root-level inports and outports

The Model Interface Package for Simulink assigns new IDs to root-level Simulink® In, Out, Bus Element In, and Bus Element Out blocks. As a result, the connections to model ports that relate to these Simulink blocks will become unresolved after updating the respective SIC files in your ConfigurationDesk application. You have to reconnect the model ports with the new IDs.

Tip

You can restore the old ID schema to maintain the connections in your existing projects by setting the new `UseClassicIdConventionForRootPorts` parameter of the `dsm_pb_pref()` API command in MATLAB to `True`:

```
dsm_pb_pref('Set', 'UseClassicIdConventionForRootPorts', true);
```

Changes to the tool automation interface that might cause code malfunctions

Some changes to the tool automation interface affect the data model and can cause code from previous Releases to malfunction. For more information, refer to [New Features and Changes to the Automation Interface for Release 2023-B \(ConfigurationDesk Automating Tool Handling !\[\]\(758ebdf4629c903da74c2e079717ae32_img.jpg\)](#)).

Migrating Projects and Applications Created with Previous Versions

Introduction

You can still open project and application files and backups created with previous Bus Manager (stand-alone) versions in the same way as you open a project from the current Bus Manager (stand-alone) version.

Note

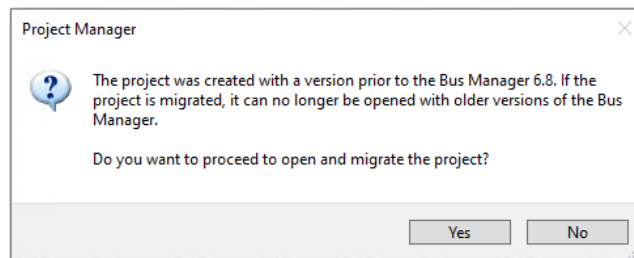
As of dSPACE Release 2021-A, the Bus Manager (stand-alone) supports the direct import only of projects last saved with one of the previous seven Bus Manager versions.

There are two different migration scenarios:

- Opening projects created with dSPACE Release 2021-A and earlier
- Opening projects created with dSPACE Release 2021-B and later

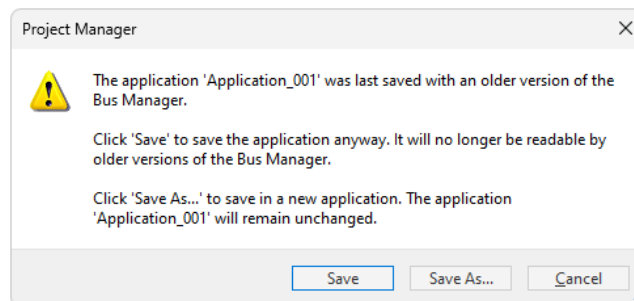
Opening projects created with dSPACE Release 2021-A and earlier

With Bus Manager (stand-alone) 6.8 (dSPACE Release 2021-B), the project structure and the project and application file formats changed. Therefore, projects created with Bus Manager (stand-alone) 6.8 and earlier cannot be used with later Bus Manager (stand-alone) versions. When you open such a project, the Bus Manager prompts you to confirm the migration as follows:



- Click Yes to migrate the project and all its applications to the new project structure and the new file formats. Afterwards, you can no longer open them with previous Bus Manager versions.
- The new project and application files are stored in the respective project and application folders. They are identified via a GUID. Do not move or rename them.

As of Bus Manager (stand-alone) 2023-A, a warning dialog is displayed before the migrated application is saved. Refer to the following example illustration.



- Click Save to migrate the application to the new Bus Manager version. Afterwards, you can no longer open the application with previous Bus Manager versions.
- Click Save As to save the application to the new Bus Manager version under a new name. The application created with the previous version will remain unchanged in the project folder.

- The migrated application files are stored in the respective application folders. They are identified via a GUID. Do not move or rename them.

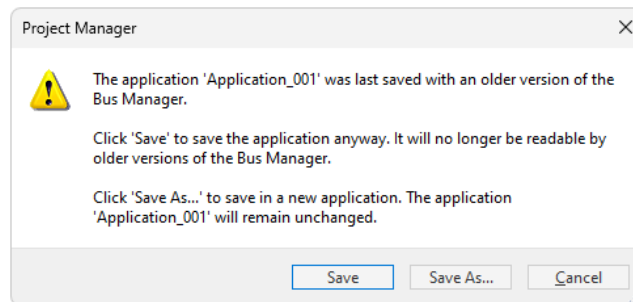
Using version control software If you are using version control software for ConfigurationDesk projects, be aware that the old project and application files are deleted and the new files might not automatically be known to your version control software when you check in your local copy.

If you checked in and deleted your local copy after the migration, you can recreate the new project and application files by checking out the project with the old project and application files again and repeating the migration. Only the project and application files will be replaced, no other migration steps will be repeated.

Opening projects created with dSPACE Release 2021-B and later

When you open a project created with Bus Manager (stand-alone) 6.8 (dSPACE Release 2021-B) and later, the Bus Manager prompts you to confirm the migration by means of a warning dialog.

As of Bus Manager (stand-alone) 2023-A, the warning dialog is displayed before the migrated application is saved and not when you open the application. Refer to the following example illustration.



- Click Save to migrate the application to the new Bus Manager version. Afterwards, you can no longer open the application with previous Bus Manager versions.
- Click Save As to save the application to the new Bus Manager version under a new name. The application created with the previous version will remain unchanged in the project folder.
- The migrated application files are stored in the respective application folders. They are identified via a GUID. Do not move or rename them.

Using version control software If you are using version control software for ConfigurationDesk projects, be aware that the old application files are deleted and the new files might not automatically be known to your version control software when you check in your local copy.

If you checked in and deleted your local copy after the migration, you can recreate the new application files by checking out the project with the old application files again and repeating the migration. Only the application files will be replaced, no other migration steps will be repeated.

Notes on the cache folder after migration

Note

Note the following specifics regarding the cache folder introduced with Bus Manager (stand-alone) 22.1 (dSPACE Release 2022-A):

- The path of project and application elements must not exceed 260 characters. Refer to [Limitations Concerning Projects and Applications \(ConfigurationDesk Real-Time Implementation Guide !\[\]\(9dc885fa0d6d341860a6e69645e59475_img.jpg\)](#)).
- Elements that are moved to the new application cache folder during project migration might exceed the character limit afterwards. To avoid this, you have to change the application cache root folder to a shorter path. For details, refer to [Project/Application Options \(ConfigurationDesk User Interface Reference !\[\]\(5d2b0686f24c91a69ec6f054f466d184_img.jpg\)](#)).

Bus Manager (Stand-Alone) Discontinuations

Discontinuation of Excel export of ConfigurationDesk applications

The Export Configuration command for exporting the configuration data of the active ConfigurationDesk application to a Microsoft® Excel™ file (XLSX file) was removed.

The corresponding `ICaAlgorithms:ExportConfiguration` tool automation API command will raise a `No more supported` exception if used.

ConfigurationDesk

Introduction With ConfigurationDesk, you can implement real-time applications for SCALEXIO hardware, MicroAutoBox III hardware, or MicroLabBox II hardware.

Where to go from here

Information in this section

New General Features of ConfigurationDesk 2023-B.....	65
New Features Concerning I/O Functionality and Hardware Support.....	67
New Features of the Bus Manager in ConfigurationDesk.....	71
Supported Container File Versions.....	74
Migrating to ConfigurationDesk 2023-B.....	77
Migrating Projects and Applications Created with Previous Versions.....	80
ConfigurationDesk Discontinuations.....	83

New General Features of ConfigurationDesk 2023-B

Enhanced FMU support

Support of FMUs providing binary input/output

variables ConfigurationDesk now supports FMUs that provide binary input/output variables according to the FMI 3.0 standard. If you use such FMUs in ConfigurationDesk, ConfigurationDesk creates a variable-size port for each binary input/output variable. You can map these ports to other variable-size model ports, or function ports that support variable-size signals.

Note

The Data vector function ports of the UDP Transmit, UDP Receive and TCP function blocks now support variable-size signals. Refer to [New Features Concerning I/O Functionality and Hardware Support](#) on page 67.

Support of FMUs providing terminals ConfigurationDesk now lets you use FMUs that provide terminals according to the FMI 3.0 and the FMI 2.0.4 standard. If you add an FMU that provides terminals with structured input or output variables to a ConfigurationDesk application, ConfigurationDesk creates a structured data inport or a structured data outport for each terminal root element. You can map these structured ports with the structured ports of other FMUs, SIC files, Simulink models, BSC files, or function blocks.

For more information on the FMU support, refer to [Working with Functional Mock-up Units \(FMU Files\) \(ConfigurationDesk Real-Time Implementation Guide !\[\]\(0f848bbd71cef6b345273b16f905912a_img.jpg\)](#)).

License check for multicore application processes

ConfigurationDesk now requires the CFD_I_MC license for application processes that use more than one processor core (Number of required processor cores ≥ 2). For more information, refer to [Overview of Licenses \(ConfigurationDesk Getting Started !\[\]\(a870788d6ed9b8fd294b7654a8c8526b_img.jpg\)](#)).

Portability of model containers between SCALEXIO and MicroLabBox II

The newly supported MicroLabBox II and SCALEXIO have the same processor architecture and Linux64-based real-time operating system. This allows you to reuse the same SIC, FMU, and BSC files in applications for both platforms.

Enhanced XCP time stamp support for ECU interfacing

ConfigurationDesk now provides enhanced XCP time-stamp support for ECU interfacing. If an ECU application with integrated XCP service supports time-stamping, the ECU Interface Configuration function block provides new TimestampOffset function ports when you import the related EIC file. The function ports provide the difference between the time of the time-stamp source and the system time of the real-time hardware. Using the TimestampOffset and the related TimestampRead function ports, you can synchronize the XCP time stamps with the system time.

For more information, refer to [Configuring the Basic Functionality \(ECU Interface Configuration\) \(ConfigurationDesk I/O Function Implementation Guide !\[\]\(6a9b39b98eb945faa14c645ec99e4eaa_img.jpg\)](#)).

New mapping file for bus accesses of bus configurations and bus simulation containers

Depending on the specific ConfigurationDesk application, ConfigurationDesk now generates EXPSWCFG.MAPPING files during the build process. The files are generated for each application process that contains CAN and/or LIN function blocks that are used to specify the bus access for bus configurations or bus simulation containers. For each CAN/LIN function block, the EXPSWCFG.MAPPING files contain the mapping to all bus access requests for which the function block specifies the bus access.

Tip

When you create a ControlDesk experiment, ControlDesk uses the `EXPSWCFG.MAPPING` files to set up the structure of the Bus Navigator tree. As a result, the Bus Navigator tree no longer differs for bus communication that is specified in bus configurations and bus simulation containers.

Improved import of FPGA containers

For FPGA container files, an `FPGA Custom Functions` folder is automatically added to the project folder to directly import an FPGA custom function to the project.

When you import an FPGA container, ConfigurationDesk now automatically replaces existing FPGA user functions with the same container ID in the selected search path. This means that only the latest version is available in the search path. The container ID is the FPGA application name that you defined in the FPGA Programming Blockset.

Redesigned Options dialog

The fully redesigned Options dialog lets you find options quickly by a category-based navigation and a search bar with direct response while typing. In addition, the Options dialog now contains the new Search Path Editor, which allows a graphical configuration of search paths for artifacts used by ConfigurationDesk (Custom Functions, FPGA containers).

For details, refer to [Options \(ConfigurationDesk User Interface Reference\)](#).

New features of the tool automation interface

The ConfigurationDesk automation interface supports additional ConfigurationDesk features. For more information, refer to [New Features and Changes to the Automation Interface for Release 2023-B \(ConfigurationDesk Automating Tool Handling\)](#).

New Features Concerning I/O Functionality and Hardware Support

New supported hardware

MicroLabBox II ConfigurationDesk supports the new MicroLabBox II.

For an overview of the function block types that support MicroLabBox II, refer to [Allocating MicroLabBox II Channel Types to Function Block Types \(ConfigurationDesk I/O Function Implementation Guide\)](#).

For information on reusing RTI models of MicroLabBox, refer to [Migrating MicroLabBox RTI Models \(MicroLabBox II - Getting Started\)](#).

SCALEXIO ConfigurationDesk supports the following new SCALEXIO hardware:

- DS6242 D/A Board

The DS6242 D/A Board is a single-slot SCALEXIO I/O board. It provides 10 galvanic-isolated analog channels for voltage out (1 channel up to 20 V,

3 channels in series up to 60 V) and current sink (1 channel up to 40 mA, 10 channels in parallel up to 400 mA).

- **DS6365-PE Automotive Ethernet Board**

The DS6365-PE Automotive Ethernet Board is a PCIe 3.0 x4 board that provides two independent ports for connecting external devices to a SCALEXIO Processing Unit via an Automotive Ethernet connection with up to 10 Gbit/s.

- **New fault simulation hardware:**

- **SCALEXIO FSX Unit**

The SCALEXIO FSX Unit is an enclosure for FSX boards. It can be built in a SCALEXIO rack and has 10 slots for FSX boards.

- **DS6801 Fault Insertion Board**

The DS6801 Fault Insertion Board is an FSX board with an IOCNET 100M communication interface. It provides 32 channels with relay-based fault insertion with up to 5 A nominal load switching current per channel.

- **DS6805 Fault Insertion Board**

The DS6805 Fault Insertion Board is an FSX board with an IOCNET 100M communication interface. It provides 10 channels with relay-based fault insertion with up to 15 A. The DS6805 also provides a central semiconductor switch for switching faults (currently not supported by software).

MicroAutoBox III ConfigurationDesk supports the new DS1521 Bus Boards, which offer Ethernet controllers with MACsec. MACsec (IEEE802.1AE) is a standard for authentication and encryption of Ethernet frames on the data link layer (layer 2).

For information on implementing MACsec, refer to [Configuring MACsec Features of Ethernet Ports \(ConfigurationDesk I/O Function Implementation Guide !\[\]\(fe3aebe81acea8d45108cd2768939da7_img.jpg\)](#)).

Enhanced function block types

UDP and TCP The Data vector function ports of the UDP Transmit, UDP Receive and TCP function blocks now support variable-size signals. The use of variable-size signals eliminates the need of an additional function port to specify the number of valid bytes within a data vector.

For more information, refer to the following topics:

- [Configuring the Basic Functionality \(TCP\) \(ConfigurationDesk I/O Function Implementation Guide !\[\]\(3292f5442e3b4027aa0bb60988f9fc82_img.jpg\)](#))
- [Configuring the Basic Functionality \(UDP Receive\) \(ConfigurationDesk I/O Function Implementation Guide !\[\]\(705a9285b25462d2e675759828e0d2ed_img.jpg\)](#))
- [Configuring the Basic Functionality \(UDP Transmit\) \(ConfigurationDesk I/O Function Implementation Guide !\[\]\(655843b308847eeb6852edac6d504d04_img.jpg\)](#))

CAN The CAN function block now lets you select the source for specifying baud rates and sample points. You can select the function block or a behavior model as the source. When you select the model, function ports are available for specifying the baud rates and sample points. In this case, you can change the baud rates and sample points at run time.








For more information, refer to [Configuring the Basic Functionality \(CAN\) \(ConfigurationDesk I/O Function Implementation Guide !\[\]\(cbd8541a32dfc32f356f5c6c994b0a21_img.jpg\)](#)).

New function block type **Angular Wavetable Current Sink** With the new Angular Wavetable Current Sink function block type, you can output a current signal sequence by looking up a table with angular-coded values.
For more information, refer to [Angular Wavetable Current Sink \(ConfigurationDesk I/O Function Implementation Guide !\[\]\(8af806fb1314382d09bc5ec5b767526c_img.jpg\)](#)).

New channel types for existing function block types The following table shows the function blocks that support new channel types and/or new hardware:

Function Block	New Channel Types	Supported Hardware	More Information
Voltage In	Analog In 23	MicroLabBox II	Voltage In (ConfigurationDesk I/O Function Implementation Guide )
	Analog In 24		
	Analog In 25		
Current Sink	Analog Out 21	DS6242 D/A Board (SCALEXIO)	Current Sink (ConfigurationDesk I/O Function Implementation Guide )
Voltage Out	Analog Out 19	MicroLabBox II	Voltage Out (ConfigurationDesk I/O Function Implementation Guide )
	Analog Out 20		
	Analog Out 21	DS6242 D/A Board (SCALEXIO)	
Multi Bit In	Digital In/Out 14	MicroLabBox II	Multi Bit In (ConfigurationDesk I/O Function Implementation Guide )
	Digital In/Out 15		
Trigger In	Digital In/Out 14	MicroLabBox II	Trigger In (ConfigurationDesk I/O Function Implementation Guide )
	Trigger In 4	DS6242 D/A Board (SCALEXIO)	
Multi Bit Out	Digital In/Out 14	MicroLabBox II	Multi Bit Out (ConfigurationDesk I/O Function Implementation Guide )
	Digital In/Out 15		
Digital Pulse Out	Digital In/Out 14	MicroLabBox II	Digital Pulse Out (ConfigurationDesk I/O Function Implementation Guide )
PWM/PFM Out	Digital In/Out 14	MicroLabBox II	PWM/PFM Out (ConfigurationDesk I/O Function Implementation Guide )
Multi-Channel PWM Out	Digital In/Out 14	MicroLabBox II	Multi-Channel PWM Out (ConfigurationDesk I/O Function Implementation Guide )
Digital Pulse Capture	Digital In/Out 14	MicroLabBox II	Digital Pulse Capture (ConfigurationDesk I/O Function Implementation Guide )
PWM/PFM In	Digital In/Out 14	MicroLabBox II	PWM/PFM In (ConfigurationDesk I/O Function Implementation Guide )
Digital Incremental Encoder In	Digital In/Out 14	MicroLabBox II	Digital Incremental Encoder In (ConfigurationDesk I/O Function Implementation Guide )
	Digital In/Out 15		
Sine Encoder In	Analog In 24	MicroLabBox II	Sine Encoder In (ConfigurationDesk I/O Function Implementation Guide )
	Analog In 25		
Hall Encoder In	Digital In/Out 14	MicroLabBox II	Hall Encoder In (ConfigurationDesk I/O Function Implementation Guide )

Function Block	New Channel Types	Supported Hardware	More Information
Resolver In	Analog In 24	MicroLabBox II	Resolver In (ConfigurationDesk I/O Function Implementation Guide 📖)
	Analog In 25		
	Analog Out 20		
EnDat Master	Digital In/Out 15	MicroLabBox II	EnDat Master (ConfigurationDesk I/O Function Implementation Guide 📖)
SSI Master	Digital In/Out 14	MicroLabBox II	SSI Master (ConfigurationDesk I/O Function Implementation Guide 📖)
	Digital In/Out 15		
Voltage Signal Capture	Analog In 23	MicroLabBox II	Voltage Signal Capture (ConfigurationDesk I/O Function Implementation Guide 📖)
	Analog In 24		
	Analog In 25		
Block-Commuted PWM Out	Digital In/Out 14	MicroLabBox II	Block-Commuted PWM Out (ConfigurationDesk I/O Function Implementation Guide 📖)
Field-Orientated Control In/Out	Analog In 23 ¹⁾	MicroLabBox II	Field-Oriented Control In/Out (ConfigurationDesk I/O Function Implementation Guide 📖)
	Analog In 24 ¹⁾		
	Analog In 25 ¹⁾		
	Digital In/Out 14 ²⁾		
Angular Clock Setup	Angle Unit Set	MicroLabBox II	Angular Clock Setup (ConfigurationDesk I/O Function Implementation Guide 📖)
SENT In	Digital In/Out 14	MicroLabBox II	SENT In (ConfigurationDesk I/O Function Implementation Guide 📖)
CAN	CAN 9	MicroLabBox II	CAN (ConfigurationDesk I/O Function Implementation Guide 📖)
LIN	LIN 5	MicroLabBox II	LIN (ConfigurationDesk I/O Function Implementation Guide 📖)
Ethernet Setup	Ethernet Adapter 1	MicroLabBox II	Ethernet Setup (ConfigurationDesk I/O Function Implementation Guide 📖)
		DS6365-PE Automotive Ethernet Board (SCALEXIO)	
Data Streaming Ethernet Setup	Ethernet Adapter 1	MicroLabBox II	Data Streaming Ethernet Setup (ConfigurationDesk I/O Function Implementation Guide 📖)
		DS6365-PE Automotive Ethernet Board (SCALEXIO)	
PTP Master	–	MicroLabBox II	PTP Master (ConfigurationDesk I/O Function Implementation Guide 📖)
		DS6365-PE Automotive Ethernet Board (SCALEXIO)	
PTP Slave	–	MicroLabBox II	PTP Slave (ConfigurationDesk I/O Function Implementation Guide 📖)

Function Block	New Channel Types	Supported Hardware	More Information
		DS6365-PE Automotive Ethernet Board (SCALEXIO)	
Shutdown Request	–	MicroLabBox II	Shutdown Request (ConfigurationDesk I/O Function Implementation Guide )
System Shutdown	–	MicroLabBox II	System Shutdown (ConfigurationDesk I/O Function Implementation Guide )
System Temperature Monitoring	–	MicroLabBox II	System Temperature Monitoring (ConfigurationDesk I/O Function Implementation Guide )
USB Eject	–	MicroLabBox II	USB Eject (ConfigurationDesk I/O Function Implementation Guide )
Domain Clock	–	MicroLabBox II	Domain Clock (ConfigurationDesk I/O Function Implementation Guide )
Non-Volatile Memory Access	–	MicroLabBox II	Non-Volatile Memory Access (ConfigurationDesk I/O Function Implementation Guide )
UART	UART 7	MicroLabBox II	UART (ConfigurationDesk I/O Function Implementation Guide )

1) Input

2) Output

New file format for hardware topologies

ConfigurationDesk can save hardware topologies to an HTFX file to export and import topologies. As of Release 2023-B, the format of the HTFX file changes from XML to JSON.

Configuration still supports the legacy XML format to be compatible with former dSPACE Releases. Select the **Hardware Topology File - Legacy Support** file type to export hardware topologies in the legacy format.

New Features of the Bus Manager in ConfigurationDesk

New supported AUTOSAR Release

The Bus Manager now supports AUTOSAR system description files based on AUTOSAR Classic Platform Release R22-11.

Transferring configured bus communication to bus configurations

The Bus Manager now lets you easily transfer configured bus communication from one bus configuration to another. To do this, you can copy entire bus configurations, or copy or cut individual bus configuration elements and paste them to one or more bus configurations in the Bus Configurations table.

For more information, refer to [Copying, Cutting, and Pasting Bus Configuration Elements \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(c444627dab9fee9a1550c053ffaaaae2_img.jpg\)](#)).

Simulating PDU gateways (preview version)

The Bus Manager now lets you simulate PDU gateways. With PDU gateways, an ECU can route the data of PDUs from one or more communication clusters to other communication clusters. PDU gateways must be specified in communication matrices. If required, you can specify user-defined PDU gateways by modifying communication matrices. However, to simulate PDU gateways, you have to enable the PDU gateway support for bus configurations.

For more information, refer to [Simulating PDU Gateways \(Preview Version\) \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(3d8c13c92b853674f749aac6fa869926_img.jpg\)](#)).

Note

The current version of simulating PDU gateways is a preview version. For more information on preview versions of Bus Manager features, refer to [Overview of the Bus Manager \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(96cc62f861fdd6e50510c0224a756dff_img.jpg\)](#)).

Generating missing ECUs for LIN PDUs

The Bus Manager now generates missing ECUs for LIN PDUs. The ECUs are generated if the communication matrix does not specify the sending or receiving ECU or if a specified ECU is not part of the communication matrix.

For more information, refer to [Working with Communication Matrices \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(e8fb589d58dad1692debababa5e928b6_img.jpg\)](#)).

Adding event-controlled timing elements to IPDUs in communication matrices

The Bus Manager now lets you add missing event-controlled timing elements to ISignal IPDUs and extended multiplexed IPDUs in communication matrices. By this, you can specify an event-controlled transmission if it is not specified yet in the communication matrix.

For more information, refer to [Adding Event-Controlled Timing Elements to IPDUs \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(d8ab143e904bfa3467271eec5af75a9b_img.jpg\)](#)).

Enhanced J1939-22 support

The Bus Manager now provides the following enhancements for J1939-22:

Manipulation of J1939-22-compliant IPDUs The Bus Manager now lets you manipulate J1939-22-compliant IPDUs with a payload length of up to 60 bytes, i.e., IPDUs that are transmitted using the Multi-PG protocol. You can assign such IPDUs and their ISignals to the Manipulation part of a bus configuration and add the PDU User Code, ISignal Overwrite Value, and ISignal Offset Value manipulation features to them.

Moreover, the Bus Manager provides the new Suspend PDU Transmission manipulation feature. The feature lets you suspend the transmission of J1939-22-compliant IPDUs that are transmitted using the Multi-PG protocol.

- For an overview of the features, refer to [Basics on Bus Configuration Features \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(73ae4d61a44e0d3e280171a702047018_img.jpg\)](#)).
- For more information on the Suspend PDU Transmission feature, refer to [Suspending the Transmission of PDUs \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(240de69c4eae84075121456f04252941_img.jpg\)](#)).

Support of assurance data The Bus Manager now supports assurance data according to the Multi-PG and Broadcast Announce Message (BAM) protocols for J1939-22-compliant IPDUs, i.e., the Bus Manager can transmit and receive assurance data.

For more information, refer to [Aspects of the J1939 Protocol \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(919a2cb85b99741a73c0c31a427236a8_img.jpg\)](#)).

Configurable communication matrix settings for J1939-22-compliant

IPDUs The Bus Manager now lets you modify the type of service, trailer format, assurance data type, and assurance data size for J1939-22-compliant IPDUs.

For more information, refer to [Configurable Settings of PDUs \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(d66ff64371a51729ac8c1cdaa685ba6f_img.jpg\)](#)).

PDU Enable feature available for multiplexed IPDUs

The Bus Manager now lets you add the PDU Enable feature to multiplexed IPDUs.

For more information, refer to [Enabling and Disabling the Transmission of PDUs \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(17413706fd4997a1a4bdf85c6864eee1_img.jpg\)](#)).

Counter Signal feature available for inspection

The Bus Manager now provides the Counter Signal feature for ISignals that are assigned to the Inspection part of a bus configuration.

For more information, refer to [Working with Counter Signals \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(d3102649f02e825ddb76dc3de0190154_img.jpg\)](#)).

Unused bit pattern of container IPDUs with a dynamic layout for padding bytes

The Bus Manager now uses the unused bit pattern of container IPDUs with a dynamic layout as the value for padding bytes of the related frames. If required, padding bytes are added to frames to extend the frame length to a valid CAN FD length.

For more information, refer to [Aspects of Miscellaneous Supported CAN Bus Features \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(b4eeff342f60cc7bcd67d869b4fedca2_img.jpg\)](#)).

Enhancements of the Restbus Configuration Exclude List

The Bus Manager now lets you add ECUs to the Restbus Configuration Exclude List via drag & drop. When you do this, the ECUs are semicolon-separated in the list and are excluded from the restbus configuration when you create it.

For more information, refer to [Creating Restbus Configurations \(ConfigurationDesk Bus Manager Implementation Guide !\[\]\(56549452e01ca28bdf2500ced9653143_img.jpg\)](#)).

Supported Container File Versions

Supported SIC file versions

ConfigurationDesk 2023-B (23.2) supports SIC file versions as listed below:

SIC Files Created With ...	MATLAB Release
dSPACE Release 2023-B: <ul style="list-style-type: none"> Model Interface Package for Simulink 2023-B (23.2) TargetLink 2023-B (23.1) 	<ul style="list-style-type: none"> R2023b R2023a R2022b R2022a
dSPACE Release 2023-A: <ul style="list-style-type: none"> Model Interface Package for Simulink 23.1 	<ul style="list-style-type: none"> R2023a R2022b R2022a R2021b
dSPACE Release 2022-B: <ul style="list-style-type: none"> Model Interface Package for Simulink 2022-B (22.2) TargetLink 5.3 	<ul style="list-style-type: none"> R2022b R2022a R2021b R2021a
dSPACE Release 2022-A: <ul style="list-style-type: none"> Model Interface Package for Simulink 22.1 	<ul style="list-style-type: none"> R2022a R2021b R2021a R2020b
dSPACE Release 2021-B: <ul style="list-style-type: none"> Model Interface Package for Simulink 4.6 TargetLink 5.2 	R2020a
dSPACE Release 2021-A: <ul style="list-style-type: none"> Model Interface Package for Simulink 4.5 	R2019b
dSPACE Release 2020-B: <ul style="list-style-type: none"> Model Interface Package for Simulink 4.4 TargetLink 5.1 	R2019a
dSPACE Release 2020-A: <ul style="list-style-type: none"> Model Interface Package for Simulink 4.3 	R2018b
dSPACE Release 2019-B: <ul style="list-style-type: none"> Model Interface Package for Simulink 4.2 	R2018a
dSPACE Release 2019-A: <ul style="list-style-type: none"> Model Interface Package for Simulink 4.1 	R2017b
dSPACE Release 2018-B: <ul style="list-style-type: none"> Model Interface Package for Simulink 4.0 	R2017a
dSPACE Release 2018-A: <ul style="list-style-type: none"> Model Interface Package for Simulink 3.6 	R2016b
dSPACE Release 2017-B: <ul style="list-style-type: none"> Model Interface Package for Simulink 3.5 	R2016a
dSPACE Release 2017-A: <ul style="list-style-type: none"> Model Interface Package for Simulink 3.4 	R2015b

Target platform compatibility of SIC files The following table shows the compatibility of SIC files and target platforms, and indicates which system target file you have to select for the generation of an SIC file:

SIC File Created With	System Target File	Target Platform
dSPACE Release 2022-A and later	dsrt.tlc	<ul style="list-style-type: none"> ▪ SCALEXIO Linux 64-bit ▪ MicroLabBox II ▪ MicroAutoBox III <p>You have to select the target architecture for which the SIC file is generated depending on the target platform. Refer to Basics on Simulink Implementation Containers (Model Interface Package for Simulink - Modeling Guide).</p>
dSPACE Release 2021-B and earlier	dsrt64.tlc (available as of dSPACE Release 2019-B)	<ul style="list-style-type: none"> ▪ SCALEXIO Linux 64-bit ▪ MicroLabBox II
	dsrt.tlc	MicroAutoBox III

Limitations for earlier SIC file versions in ConfigurationDesk scenarios

SIC files created with the Model Interface Package for Simulink version 3.4 ... 4.1 are not supported in the following ConfigurationDesk scenarios:

- In multimodel application processes.
- For building real-time applications that use Real-Time Testing.

Limitations for Simulink behavior models underlying earlier SIC file versions

The following limitations apply to Simulink behavior models underlying SIC files created with the Model Interface Package for Simulink version 3.4 ... 4.4:

- The Simulink behavior model must not contain blocks from the following blocksets:
 - Blocks of ASM
 - Real-time applications that contain such SIC files cannot be used with ModelDesk.
 - FPGA Programming Blockset
 - MotionDesk Blockset
 - Real-time applications that contain such SIC files cannot be used with MotionDesk.
 - Blocks of any dSPACE Solution.
- Compiled objects contained in SIC files must match the target platform. SIC files with compiled objects that were created at a time when the current target platforms did not exist are not supported. In this case, it is not possible to use compiled objects.

Supported BSC file versions

ConfigurationDesk 2023-B (23.2) supports BSC files that were generated with the Bus Manager of the current Release, i.e., BSC file version 1.12, or with the Ethernet Configuration Package. For more information on supported Ethernet BSC file versions, refer to <http://www.dspace.com/go/ethcpinfo>.

Supported Functional Mock-up Unit versions

ConfigurationDesk 2023-B (23.2) supports Functional Mock-up Units (FMUs) that comply with the following versions of the FMI standard:

- 2.0.x
- 3.0.x

Supported EIC file versions

ConfigurationDesk 2023-B (23.2) supports EIC file versions as listed below:

EIC Files Created With ...	EIC Version
dSPACE Release 2023-B (ECU Interface Manager 2023-B)	5.0.0
dSPACE Release 2023-A (ECU Interface Manager 2023-A)	5.0.0
dSPACE Release 2022-B (ECU Interface Manager 2022-B)	5.0.0
dSPACE Release 2022-A (ECU Interface Manager 2.11)	5.0.0
dSPACE Release 2021-B (ECU Interface Manager 2.10)	4.0.0
dSPACE Release 2021-A (ECU Interface Manager 2.9)	4.0.0
dSPACE Release 2020-B (ECU Interface Manager 2.8)	4.0.0
dSPACE Release 2020-A (ECU Interface Manager 2.7)	4.0.0
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

Note

- For 64-bit target architectures, only EIC files as of version 5.0.0 are supported.
- For MicroAutoBox III systems, only EIC files as of version 4.0.0 are supported.

Migrating to ConfigurationDesk 2023-B

Installation path changes

As of dSPACE Release 2023-A, ConfigurationDesk is installed in a separate folder named `dSPACE ConfigurationDesk <ReleaseVersion>`. This means ConfigurationDesk is no longer part of the RCP and HIL installation folder.

If you have scripts depending on the ConfigurationDesk installation location, you need to adapt those scripts.

Changed order in TRC files and A2L files

Groups, blocks, and variables in TRC files and A2L files are now sorted alphabetically. As a consequence, you might have to adapt your custom tools based on these files.

Migrating to new ID behavior of Simulink root-level inports and outports

The Model Interface Package for Simulink assigns new IDs to root-level Simulink® In, Out, Bus Element In, and Bus Element Out blocks. As a result, all connections to model ports in your ConfigurationDesk or VEOS projects, which relate to these Simulink blocks, will become unresolved after analyzing the model or updating the respective SIC file. You have to reconnect the model ports with the new IDs.

Tip

You can restore the old ID schema to maintain the connections in your existing projects by setting the new `UseClassicIdConventionForRootPorts` parameter of the `dmpb_pref()` API command in MATLAB to `True`:

```
dmpb_pref('Set', 'UseClassicIdConventionForRootPorts', true);
```

Changes to the target architecture settings

The support of the SCALEXIO 32-bit Linux target platform was removed with ConfigurationDesk 2023-B (23.2). This results in the following changes:

- The SCALEXIO target architecture property for specifying the default target architecture was removed from the Configuration page of the Options dialog.
- The Target architecture property in the Global Build Settings is now read-only. Depending on the assigned processing unit, one of the following values is displayed:
 - 64-bit (for SCALEXIO and MicroLabBox II)
 - 32-bit (for MicroAutoBox III)
 - Not applicable (if no processing unit is assigned)
- The SCALEXIO_LNX target platform identifier is no longer supported for precompiling model containers and in automation commands.

The SCALEXIO 64-bit Linux target platform (SCALEXIO_LNX64) is still available.

Note

Model implementation containers that include 32-bit-compatible binaries are not compatible with the SCALEXIO Linux or MicroLabBox II 64-bit target architecture. You have to generate the model implementation container again with Linux 64-bit-compatible binaries.

SIC files must be generated again with the appropriate Target architecture setting in the Simulink Configuration Parameters dialog or via the `dsrt_build()` API command.

Changes to the tool automation interface that might cause code malfunctions

Some changes to the tool automation interface affect the data model and can cause code from previous Releases to malfunction. For more information, refer to [New Features and Changes to the Automation Interface for Release 2023-B \(ConfigurationDesk Automating Tool Handling !\[\]\(5a132f13505a6571904d622757b7a8f0_img.jpg\)](#)).

Changes to custom function folders

As of dSPACE Release 2022-A, ConfigurationDesk does not automatically add the project-specific custom function folder `<ProjectLocation>\<ProjectFolder>\CustomFunctions` to a new project. The lack of the project-specific custom function folder might affect the execution of automation scripts that copy custom functions to that folder.

For new projects, it is recommended that you use search paths to add custom functions. Search paths let you use repositories and version control software, for example. For more information, refer to [Basics on Search Paths and User Location Files \(ConfigurationDesk Custom I/O Function Implementation Guide !\[\]\(73002692dd5e7a64e60946be3158e719_img.jpg\)](#)).

However, ConfigurationDesk still supports project-specific custom function folders.

Changed configuration settings for generating digital output signals

With dSPACE Release 2023-A, the settings for the High reference potential property have changed. If you open a project that was created with a previous ConfigurationDesk version, the settings will be migrated as shown in the following table.

For each relevant channel type, the table shows the settings before migration, and the new mapped settings after migration. The settings are shown with their integer values for automation purposes (in parentheses).

Note

You have to check whether the new mapped settings are applicable. If necessary, you must change the settings according to your requirements.

Old settings Channel Type	Individual (1)	VBat (2)	Shared (4)	Shared 2 (5)
No hardware resource assignment	External individual (1)	External shared (2)	External shared (2)	External shared 2 (3)
Flexible Out 1	External individual (1)	External individual (1)	External individual (1)	External individual (1)
Digital Out 1	External individual (1)	Internal shared (4)	External shared (2)	External shared (2)
Digital Out 2	External individual (1)	External individual (1)	External individual (1)	External individual (1)
Digital Out 3	External individual (1)	External shared (2)	External shared (2)	External shared 2 (3)
Digital Out 4	External individual (1)	External shared (2)	External shared (2)	External shared (2)
Digital Out 5	External individual (1)	External shared (2)	External shared (2)	External shared (2)
Digital Out 7	External individual (1)	External shared (2)	External shared (2)	External shared (2)
Digital Out 8	External individual (1)	External shared (2)	Internal shared (4)	Internal shared 2 (5)
Digital In/Out 1	External individual (1)	External individual (1)	External individual (1)	External individual (1)
Digital In/Out 3	External individual (1)	External shared (2)	External shared (2)	External shared 2 (3)
Digital In/Out 5	External individual (1)	External shared (2)	Internal shared (4)	Internal shared 2 (5)
Digital In/Out 6	External individual (1)	External shared (2)	Internal shared (4)	Internal shared (4)
Digital In/Out 8	External individual (1)	External shared (2)	Internal shared (4)	Internal shared (4)
Digital In/Out 9	External individual (1)	External shared (2)	Internal shared (4)	Internal shared 2 (5)
Digital In/Out 10	External individual (1)	Internal shared 2 (5)	Internal shared (4)	Internal shared 2 (5)

A conflict is generated and displayed if the channel type does not support the selected setting. For example, the Flexible Out 1 channel type supports only the External individual setting. For all other settings, a conflict is generated.

Description of the new settings

Setting	Supported Channel Types	Description
External individual	<ul style="list-style-type: none"> Flexible Out 1 (on DS2621) 	<p>The channels of Flexible Out 1 channel type are galvanically isolated to the ground potential of the dSPACE real-time hardware. Each channel provides its own electrical connection and is independent of the other channels. This allows the connection of a potential shifted signal.</p> <p>You can connect any potential to the High Reference signal port.</p>
	<ul style="list-style-type: none"> Digital Out 2 (on DS2690) Digital In/Out 1 (on DS2690) 	<p>Each channel of the channel types provides its own electrical high reference potential. Therefore, each channel is independent of the other channels. This allows the connection of an individual arbitrary potential.</p> <p>You can connect any potential to the High Reference signal port.</p>
External shared	<ul style="list-style-type: none"> Digital Out 1 (on DS2680) Digital Out 3 (on DS6101) Digital Out 4 (on MircoAutoBox III) Digital Out 5 (on MircoAutoBox III) 	<p>These channel types share the connected potential with all other digital output channels that belong to a channel set. The potential is therefore available on a single pin of the I/O connector.</p> <p>You can connect any potential to the High Reference signal port.</p>

Setting	Supported Channel Types	Description
	<ul style="list-style-type: none"> ▪ Digital Out 7 (on MircoAutoBox III) ▪ Digital In/Out 3 (on DS6201) 	
External shared 2	<ul style="list-style-type: none"> ▪ Digital Out 3 (on DS6101) ▪ Digital In/Out 3 (on DS6201) 	These channel types share the connected potential with all other digital output channels that belong to a channel set. The potential is therefore available on a single pin of the I/O connector. The connector pin that shares this potential (External shared 2) is not the same as the pin that is used for the External shared setting. You can connect any potential to the High Reference signal port.
Internal shared	<ul style="list-style-type: none"> ▪ Digital Out 8 (on DS6121) ▪ Digital In/Out 5 (on DS6202) ▪ Digital In/Out 6 (on MircoAutoBox III) ▪ Digital In/Out 8 (on MircoAutoBox III) ▪ Digital In/Out 9 (on DS6121) ▪ Digital In/Out 10 (on MircoAutoBox III) 	The internal 5 V supply of the dSPACE real-time hardware is used as high-reference potential. The High Reference signal port is not available.
	<ul style="list-style-type: none"> ▪ Digital Out 1 (on DS2680) 	The internal VBAT (simulated battery voltage) of the dSPACE real-time system is used as high-reference potential. The High Reference signal port is not available. The simulated battery voltage of a SCALEXIO system is provided by the battery simulation power supply unit connected to the DS2680. The voltage range depends on the power supply used. Refer to Battery Simulation Power Supply Unit (SCALEXIO Hardware Installation and Configuration) .
Internal shared 2	<ul style="list-style-type: none"> ▪ Digital Out 8 (on DS6121) ▪ Digital In/Out 5 (on DS6202) ▪ Digital In/Out 9 (on DS6121) 	The internal 3.3 V supply of the dSPACE real-time hardware is used as high-reference potential. The High Reference signal port is not available.
	<ul style="list-style-type: none"> ▪ Digital In/Out 10 (on MircoAutoBox III) 	The internal VBAT of the dSPACE real-time hardware is used as high-reference potential. The High Reference signal port is not available. The internal VBAT of the MicroAutoBox III corresponds to the operating voltage of the MicroAutoBox III. For values, refer to General Characteristics (MicroAutoBox III Hardware Installation and Configuration) .

Migrating Projects and Applications Created with Previous Versions

Introduction

You can still open project and application files and backups created with previous ConfigurationDesk versions in the same way as you open a project from the current ConfigurationDesk version.

Note

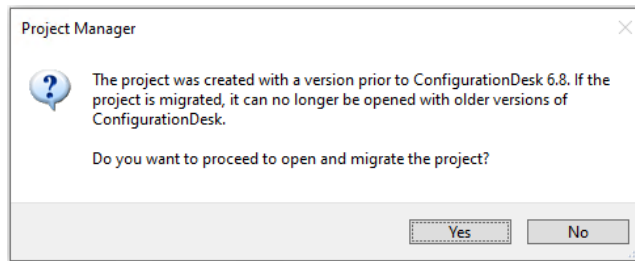
As of dSPACE Release 2021-A, ConfigurationDesk supports the direct import only of projects last saved with one of the previous seven ConfigurationDesk versions.

There are two different migration scenarios:

- Opening projects created with dSPACE Release 2021-A and earlier
- Opening projects created with dSPACE Release 2021-B and later

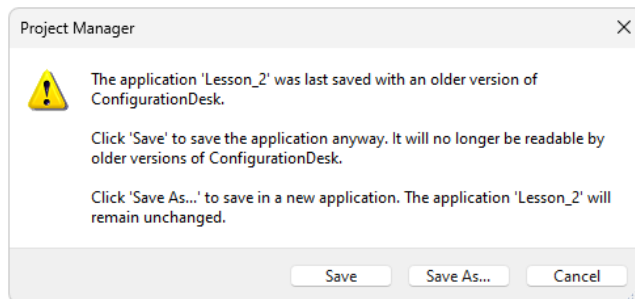
Opening projects created with dSPACE Release 2021-A and earlier

With ConfigurationDesk 6.8 (dSPACE Release 2021-B) the project structure, project and application file formats changed. Therefore, projects created with ConfigurationDesk 6.8 and earlier cannot be used with later ConfigurationDesks versions. When you open such a project, ConfigurationDesk prompts you to confirm the migration as follows:



- Click Yes to migrate the project and all its applications to the new project structure and the new file formats. Afterwards, you can no longer open them with previous ConfigurationDesk versions.
- The new project and application files are stored in the respective project and application folders. They are identified via a GUID. Do not move or rename them.

As of ConfigurationDesk 2023-A, a warning dialog is displayed before the migrated ConfigurationDesk application is saved. Refer to the following example illustration.



- Click Save to migrate the application to the new ConfigurationDesk version. Afterwards, you can no longer open the application with previous ConfigurationDesk versions.
- Click Save As to save the application to the new ConfigurationDesk version under a new name. The application created with the previous version will remain unchanged in the project folder.

- The migrated application files are stored in the respective application folders. They are identified via a GUID. Do not move or rename them.

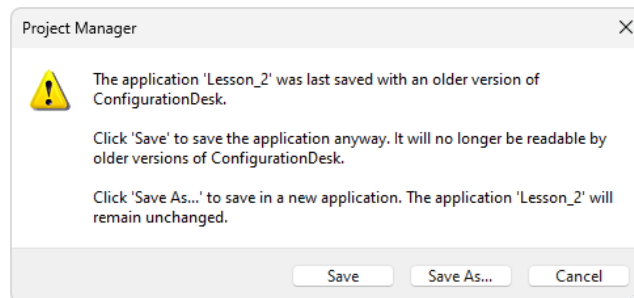
Using a version control software If you are using version control software for ConfigurationDesk projects, be aware that the old project and application files are deleted and the new files might not automatically be known to your version control software when you check in your local copy.

If you checked in and deleted your local copy after the migration, you can recreate the new project and application files by checking out the project with the old project and application files again and repeating the migration. Only the project and application files will be replaced, no other migration steps will be repeated.

Opening projects created with dSPACE Release 2021-B and later

When you open a project created with ConfigurationDesk 6.8 (dSPACE Release 2021-B) and later, ConfigurationDesk prompts you to confirm the migration by means of a warning dialog.

As of ConfigurationDesk 2023-A, the warning dialog now is displayed before the migrated ConfigurationDesk application is saved and not when you open the application. Refer to the following example illustration.



- Click Save to migrate the application to the new ConfigurationDesk version. Afterwards, you can no longer open the application with previous ConfigurationDesk versions.
- Click Save As to save the application to the new ConfigurationDesk version under a new name. The application created with the previous version will remain unchanged in the project folder.
- The migrated application files are stored in the respective application folders. They are identified via a GUID. Do not move or rename them.

Using a version control software If you are using version control software for ConfigurationDesk projects, be aware that the old application files are deleted and the new files might not automatically be known to your version control software when you check in your local copy.

If you checked in and deleted your local copy after the migration, you can recreate the new application files by checking out the project with the old application files again and repeating the migration. Only the application files will be replaced, no other migration steps will be repeated.

Notes on regarding the cache folder after migration

Note

Note the following specifics regarding the cache folder introduced with ConfigurationDesk 22.1 (dSPACE Release 2022-A):

- The path of project and application elements must not exceed 260 characters. Refer to [Limitations Concerning Projects and Applications \(ConfigurationDesk Real-Time Implementation Guide !\[\]\(aca6fcc8bd95e8255b9ea1b1d08ef300_img.jpg\)](#)).
- Elements such as build artifacts that are moved to the new application cache folder during project migration might exceed the character limit afterwards. To avoid this, you have to change the application cache root folder to a shorter path. For details, refer to [Project/Application Options \(ConfigurationDesk User Interface Reference !\[\]\(0083087c61cec498ac803a4aec5bb1bd_img.jpg\)](#)).

ConfigurationDesk Discontinuations

Discontinuation of SCALEXIO 32-bit Linux target platform

The support of the SCALEXIO 32-bit Linux target platform was removed with ConfigurationDesk 2023-B (23.2). For migration notes concerning the target architecture settings, refer to [Migrating to ConfigurationDesk 2023-B](#) on page 77.

Discontinuation of Excel export of ConfigurationDesk applications

The Export Configuration command for exporting the configuration data of the active ConfigurationDesk application to a Microsoft® Excel™ file (XLSX file) was removed with ConfigurationDesk 2023-B (23.2).

The corresponding tool automation API command `ICaAlgorithms:ExportConfiguration` will raise a `No more supported` exception if used.

Discontinuation of XLS files containing device topologies

With ConfigurationDesk 2023-A (23.1), ConfigurationDesk lets you import external device topologies only from DTFX and Microsoft Excel™ XLSX files.

Discontinuation of custom device properties

The possibility to add custom device properties to device topology elements was removed with ConfigurationDesk 2023-B (23.2).

Planned discontinuation of external cable harness calculation

The possibility to calculate the representation of an external cable harness will be removed with ConfigurationDesk 2024-A.

Planned discontinuation of RTI bus communication blockset support for SCALEXIO

The support of the RTI CAN MultiMessage Blockset and the RTI LIN MultiMessage Blockset for SCALEXIO will be removed with ConfigurationDesk 2025-A. We recommend not to use models that contain blocks from these blocksets in new ConfigurationDesk projects for SCALEXIO.

ControlDesk

Where to go from here

Information in this section

New Features of ControlDesk 2023-B.....	86
Migrating to ControlDesk 2023-B.....	102

New Features of ControlDesk 2023-B

Where to go from here

Information in this section

New Features of Platform Management and Platforms/Devices (ControlDesk 2023-B)	86
Provides an overview of new features of platform management and platforms/devices in ControlDesk 2023-B.	
New Variable Management Features (ControlDesk 2023-B)	87
Provides an overview of the new variable management features in ControlDesk 2023-B.	
New Instrument Features (ControlDesk 2023-B)	88
Provides an overview of the new instrument features in ControlDesk 2023-B.	
New Bus Navigator Features (ControlDesk 2023-B)	91
Provides an overview of the new Bus Navigator features in ControlDesk 2023-B.	
New Electrical Error Simulation Features (ControlDesk 2023-B)	99
Provides an overview of the new electrical error simulation features as of ControlDesk 2023-B.	
New User Interface Handling Features (ControlDesk 2023-B)	99
Provides an overview of the new user interface handling features in ControlDesk 2023-B.	

New Features of Platform Management and Platforms/Devices (ControlDesk 2023-B)

Platform support for the new MicroLabBox II

ControlDesk features the MicroLabBox II platform to support the new MicroLabBox II.

Refer to [MicroLabBox II Platform \(ControlDesk Platform Management !\[\]\(e1d6102fe77919492c04879c8450f1f5_img.jpg\)](#)).

SCALEXIO platform: Support of the DS6242 D/A Board and the DS6365-PE Automotive Ethernet Board

ControlDesk now supports the following new SCALEXIO boards:

- DS6242 D/A Board
- DS6365-PE Automotive Ethernet Board

GNSS device: New status variables, e.g., to indicate the number of satellites in use

As of ControlDesk 2023-B, the GNSS device provides the Status variable group as shown in the following illustration:

Group	Description	Favorite	Var Co	Variable	Block	Platform/Device	Description	Unit	Type
All Variable Descriptions		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Quality	Status	GNSS	Quality of the satellite fix.		Signed 64 bit
Calculated Variables		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Satellites	Status	GNSS	Number of satellites in use.		Signed 64 bit
GNSS Data		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Status	Status	GNSS	Status of the receiver.		Signed 64 bit
Date									
Position									
Speed									
Status									
Time									
VEOS_demo.sdf									

Variable	Purpose
Quality	To display the current satellite fix, i.e., the type of signal or technique being used by the GNSS receiver to determine its location. The satellite fix therefore indicates the quality of the signal, or the accuracy and reliability of the location being reported.
Satellites	To display the number of satellites that the GNSS receiver currently uses.
Status	To display whether the connected GNSS receiver receives valid data.

Exporting the hardware topology

For the MicroAutoBox III, MicroLabBox II, and SCALEXIO platforms, ControlDesk now lets you export the hardware topology for reuse in ConfigurationDesk applications.

Refer to [Export Hardware Topology \(ControlDesk Platform Management !\[\]\(642aa997563f9a325b310230bb5078b7_img.jpg\)](#)).


New Variable Management Features (ControlDesk 2023-B)

Variable mapping improvements

The ControlDesk support for variable mapping according to *identifier mapping of the ASAM XIL Framework mapping concept*, which was introduced with Version 2023-A, was improved:

Replacing mappings with variables on layouts, and vice versa On layouts, you can now replace resolvable mappings connected to instruments with their associated variables, and vice versa. You can do so separately for the active layout, for all open layouts, and for all the layouts of the experiment.

Refer to [Replace Variables with Mappings \(ControlDesk Layouting !\[\]\(c444627dab9fee9a1550c053ffaaaae2_img.jpg\)](#)), [Replace Mappings with Variables \(ControlDesk Layouting !\[\]\(e4a71fb14267cbc3c68a54ad33289c8f_img.jpg\)](#)) and [LayoutManagement / IXLayoutManagement <<Interface>> \(ControlDesk Automation !\[\]\(14c85d5bb83aa7451202bf95a5e535fd_img.jpg\)](#)).

RT application demos use variable mapping The ControlDesk real-time application demo projects now use variable mapping. The layouts and instruments of the demo experiments are based on framework labels. Refer to [Real-Time Application Demos \(ControlDesk Introduction and Overview\)](#) .

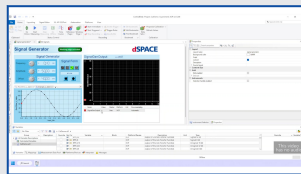
New Instrument Features (ControlDesk 2023-B)

Time Plotter, Index Plotter: Displaying signal names in the legend

ControlDesk now lets you display *signal names* in the Time Plotter and Index Plotter legend.

Time Plotter: Displaying Signal Names in the Legend

This video shows you how to display signal names in the Time Plotter legend. The procedure for the Index Plotter is the same.



To watch this video, click the following link or scan the QR code:
<https://www.dspace.com/dspace-help/zdaqb>

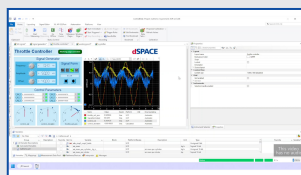
Refer to [Column Chooser \(Time Plotter/Index Plotter\)](#) (ControlDesk Instrument Handling .

Time Plotter: Showing the connection name in time and data cursor tooltips

ControlDesk now lets you show *the name of the connected variable* in the time and data cursor tooltips of the Time Plotter.

Time Plotter: Showing the Connection Name in Time and Data Cursor Tooltips

This video shows you how to enable the display of the connection name in the time and data cursor tooltips of the Time Plotter.



To watch this video, click the following link or scan the QR code:
<https://www.dspace.com/dspace-help/T9Nrk>

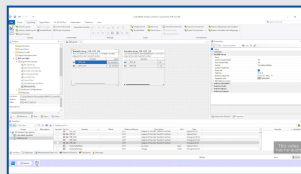
Refer to [Time Cursor Properties \(ControlDesk Instrument Handling !\[\]\(c8d96c8885d3000a912c2582004aed63_img.jpg\)\)](#) and [Data Cursor Properties \(Time Plotter/Index Plotter\) \(ControlDesk Instrument Handling !\[\]\(3ad821e3ca7dd4cb7003e9c8d982e254_img.jpg\)\)](#).

Variable Array: Specifying custom header text

ControlDesk now lets you specify *custom text* for the header of Variable Arrays. This is useful, for example, to split a group of variables across multiple Variable Arrays and use the group name for all the instruments.

Variable Array: Specifying Custom Header Text

This video shows you how to specify custom header text for the Variable Array via the Properties pane and via the `IViVariableArrayInstrument` automation interface.



To watch this video, click the following link or scan the QR code:
<https://www.dspace.com/dspace-help/d6gcM>

Refer to [Custom Header Text Property \(Variable Array\) \(ControlDesk Instrument Handling !\[\]\(faf942dc3e59ce8eb64b4ac481eca7e0_img.jpg\)\)](#) and [VariableArrayInstrument / IViVariableArrayInstrument <<Interface>> \(ControlDesk Automation !\[\]\(f6b0299e0b5e4340e509b71914970da0_img.jpg\)\)](#).

Sound Controller: MP3 file support

The ControlDesk Sound Controller now also supports MP3 sound files.

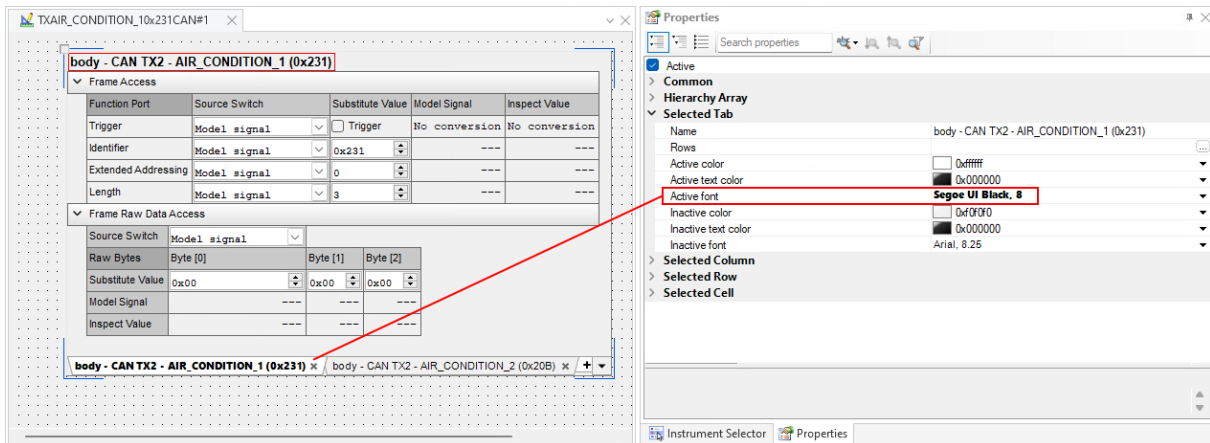
Refer to [Sound Controller Properties \(ControlDesk Instrument Handling !\[\]\(4b7a79268f6ba26c1471d4232fffa85a_img.jpg\)\)](#) and [SoundController / IViSoundController <<Interface>> \(ControlDesk Automation !\[\]\(87d978583253c9bde1db2d6dfafe8de0_img.jpg\)\)](#).

Hierarchy Array: Usability improvements

The Hierarchy Array was improved.

Customizing the font of tab header text You can now customize the font of the text in the tab header. You can customize the font separately for the headers of the active and inactive tabs.

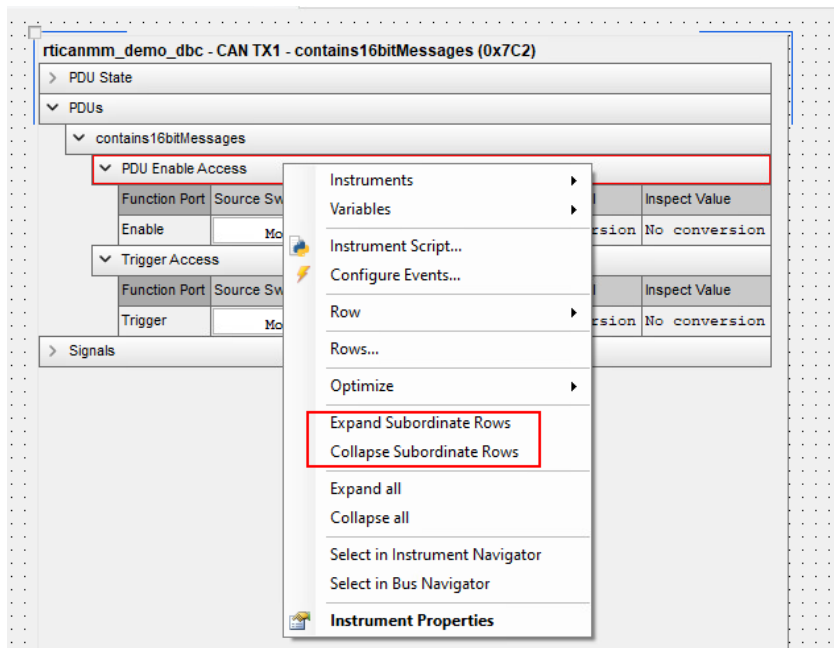
The following illustration shows a Hierarchy Array with two tabs. The header text font of the currently active tab is customized.



Refer to [Tabs Properties \(Hierarchy Array\) \(ControlDesk Instrument Handling\)](#).

Expanding/collapsing subordinate rows You can now expand/collapse all the subordinate rows of a specific header row.

The following illustration shows the related commands in the instrument context menu. For an animated graphic, refer to dSPACE Help.



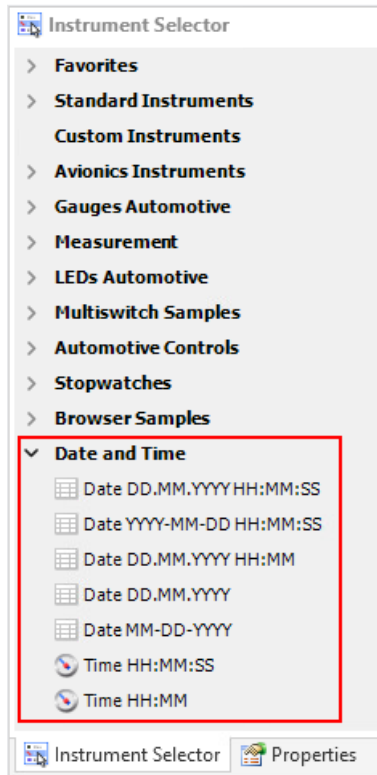
Moving the instrument by dragging the tab header You can now move the Hierarchy Array by dragging the tab header.

For an animated graphic, refer to dSPACE Help.

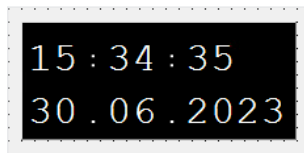
Date and Time instrument category

The ControlDesk Instrument Selector now provides the Date and Time category with preconfigured instruments that let you display the host PC system time and/or date on a layout in different formats.

The following illustration shows the instrument category:



The following illustration shows the Date DD.MM.YYYY HH:MM:SS instrument as an example:



Refer to [Date and Time Library \(ControlDesk Instrument Handling\)](#).

New Bus Navigator Features (ControlDesk 2023-B)

Usage of port information from the AUTOSAR file for decoding SOME/IP-SD messages

Up to and including ControlDesk 2023-A, specifying related ports in the Manage SOME/IP-SD Ports dialog was mandatory to decode SOME/IP-SD messages.

As of ControlDesk 2023-B, this is no longer mandatory because ControlDesk can extract the required information from the AUTOSAR variable description file of the related Ethernet Bus Monitoring device.

However, if the AUTOSAR variable description file contains no matching port information, you can use the Provide Fallback Ports for SOME/IP dialog to specify fallback ports.

For more information, refer to [Manage SOME/IP-SD Ports \(ControlDesk Bus Navigator !\[\]\(21199eb166cc97331a0c54c649195dcc_img.jpg\)\)](#).

Importing and visualizing SOME/IP event parameters (preview feature)

As of ControlDesk 2023-B, ControlDesk can import SOME/IP event parameters from the AUTOSAR variable description file of the related Ethernet Bus Monitoring device. You can then visualize the parameters on a layout, and view the decoded event parameter in the Decoding View.

Note

Importing and using SOME/IP event parameters in ControlDesk 2023-B is a *preview feature*. Preview features are features in an early but completely tested development stage. They might benefit from user feedback before being finally integrated into the regular product.

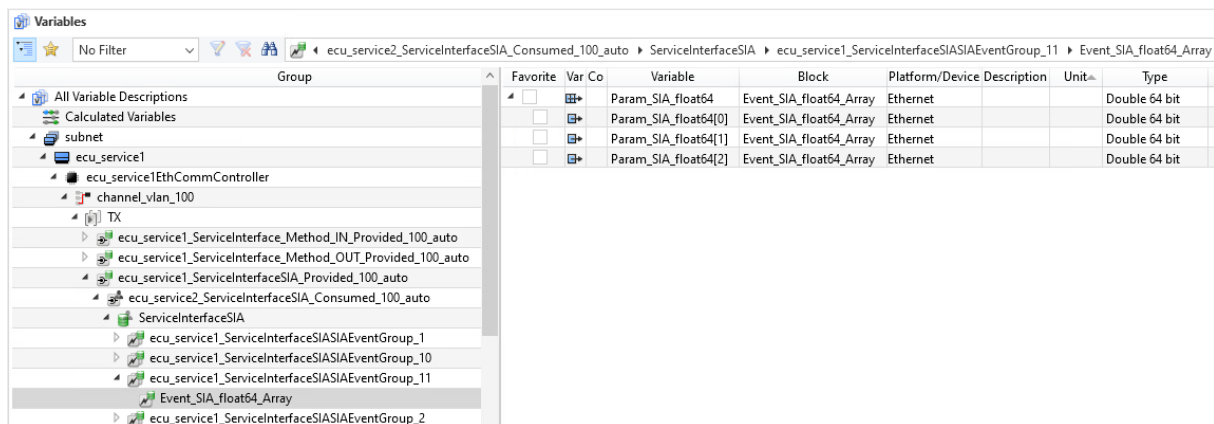
To provide feedback on or to request a feature, or report a problem, contact us by sending an e-mail to support@dspace.de.

To import SOME/IP event parameters and visualize them, perform the following steps:

1. Enable the import of SOME/IP event parameters on the Preview Features page.
2. Add an Ethernet Bus Monitoring device to the experiment, and add the AUTOSAR variable description file to the device.

If your experiment already contains an Ethernet Bus Monitoring device, reload the AUTOSAR variable description file.

The Variables pane displays the imported SOME/IP event parameters. The following illustration shows an example:



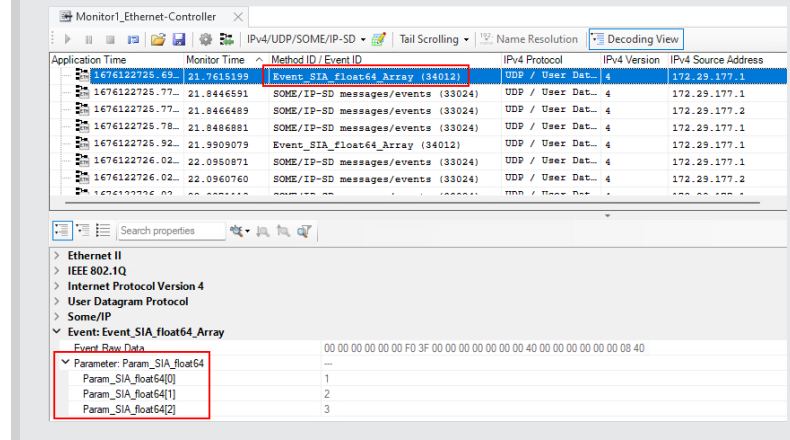
The screenshot shows the 'Variables' pane in ControlDesk. The left sidebar displays a tree view of variable descriptions, with the path 'ecu_service2_ServiceInterfaceSIA_Consumed_100_auto' expanded to show 'ServiceInterfaceSIA' and 'Event_SIA_float64_Array'. The main pane displays a table of variables:

Favorite	Var Co	Variable	Block	Platform/Device Description	Unit	Type
<input type="checkbox"/>		Param_SIA_float64	Event_SIA_float64_Array	Ethernet		Double 64 bit
<input type="checkbox"/>		Param_SIA_float64[0]	Event_SIA_float64_Array	Ethernet		Double 64 bit
<input type="checkbox"/>		Param_SIA_float64[1]	Event_SIA_float64_Array	Ethernet		Double 64 bit
<input type="checkbox"/>		Param_SIA_float64[2]	Event_SIA_float64_Array	Ethernet		Double 64 bit

- Visualize the SOME/IP event parameters on a layout according to your needs.

Tip

The Decoding View also displays the decoded SOME/IP event parameters. The following illustration shows an example:



Note

The following limitations apply:

- Only SOME/IP event parameters with a basic data type and one-dimensional arrays with fixed length are supported.
- SOME/IP event parameters are not displayed in the Bus Navigator tree.

Customizing Bus Instrument generation (preview feature)

ControlDesk 2023-B lets you customize the generation of Bus Instruments that are based on the Hierarchy Array, i.e., Bus Instruments created for bus communication modeled with the Bus Manager.

Note

The customization of Bus instrument generation in ControlDesk 2023-B is a *preview feature*. Preview features are features in an early but completely tested development stage. They might benefit from user feedback before being finally integrated into the regular product.

To provide feedback on or to request a feature, or report a problem, contact us by sending an e-mail to support@dSPACE.de.

To create custom Bus Instruments, you have to perform the following steps:

- Create a separate layout description file for each type of custom Bus Instrument.
- Enable the usage of custom layout description files on the Preview Features page of the ControlDesk Options dialog.
- Specify the folder that contains your custom layout description files on the Bus Navigator - Layout Generation page of the ControlDesk Options dialog.

For more information, refer to [Creating Custom Bus Instruments \(ControlDesk Bus Navigator !\[\]\(d0a1791f26d167e866e44ebbf83efebe_img.jpg\)\)](#).

J1939-22: Generating Compact and Manipulation Bus Instruments for Multi-PG IPDUs (Bus Manager applications only)

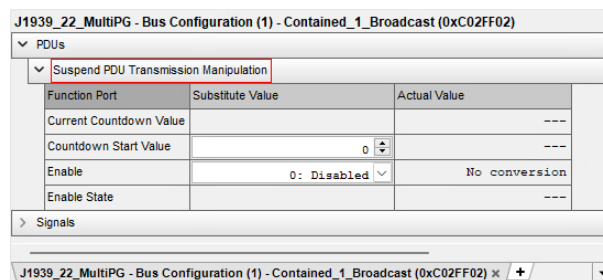
As of ControlDesk 2023-B, you can generate Compact and Manipulation Bus Instruments for J1939-22-compliant IPDUs with a payload length of up to 60 bytes, i.e., IPDUs that are transmitted using the Multi-PG protocol.

For more information, refer to:

- [Compact Bus Instrument \(Bus Manager\) \(ControlDesk Bus Navigator !\[\]\(9bf097d682561b2ffd12d57a40ca73b1_img.jpg\)\)](#)
- [Manipulation Instrument for CAN \(Bus Manager\) \(ControlDesk Bus Navigator !\[\]\(51d3868eac81c232f6ef399d2bd16077_img.jpg\)\)](#)

J1939-22: Suspend PDU transmission manipulation support (Bus Manager applications only)

As of ControlDesk 2023-B, the Compact Bus Instrument and the Manipulation Instrument for CAN generated for bus communication modeled with the Bus Manager support the *Suspend PDU transmission manipulation* feature by displaying the new Suspend PDU Transmission Manipulation region. The region lets you suspend the transmission of a J1939-22 IPDU.



For more information, refer to:

- [Compact Bus Instrument \(Bus Manager\) \(ControlDesk Bus Navigator !\[\]\(3168ddc4389f6b417dd71f084513be9c_img.jpg\)\)](#)
- [Manipulation Instrument for CAN \(Bus Manager\) \(ControlDesk Bus Navigator !\[\]\(17332056424eb04f01463711418ba65a_img.jpg\)\)](#)

PDU user port manipulation support (Bus Manager applications only)

As of ControlDesk 2023-B, the Manipulation Instrument for CAN generated for bus communication modeled with the Bus Manager supports the *PDU user port manipulation* feature by displaying the user ports in the PDU User Code Manipulation region. The region lets you configure the values of the user ports.

CanCluster - BusConfig1 - IPduA1 (0x051)

▼ PDUs

▼ PDU User Code Manipulation

Function Port	Substitute Value	Actual Value
Current Countdown Value		---
Countdown Start Value	0	---
Enable	0: Disabled	No conversion
Enable State		---
User Inport 0	0	---
User Inport 1	0	---
User Output 2		---
User Inport 3	0	---
User Inport 4	0	---
User Output 5		---
User Output 6		---
User Inport 7	0	---
User Inport 8	0	---
User Output 9		---
User Inport 10	0	---
User Inport 11	0	---
User Inport 12	0	---
User Inport 13	0	---
User Inport 14	0	---
User Inport 15	0	---
User Inport 16	0	---
User Inport 17	0	---
User Inport 18	0	---
User Inport 19	0	---

CanCluster - BusConfig1 - IPduA1 (0x051) x +

For more information, refer to [Manipulation Instrument for CAN \(Bus Manager\) \(ControlDesk Bus Navigator !\[\]\(d84e7ea36f695d92cb39ec32c307ac93_img.jpg\)](#)).

SecOC freshness overwrite value manipulation support (Bus Manager applications only)

As of ControlDesk 2023-B, the Manipulation Instrument for CAN generated for bus communication modeled with the Bus Manager supports the *SecOC freshness overwrite value manipulation* feature by displaying the new SecOC Freshness Overwrite Value Manipulation region. The region lets you overwrite the freshness value of the selected secured IPDU.

CanCluster - BusConfiguration_TX - IPdu02_FVCounter_secured (0x011)

▼ PDUs

> PDU SecOC Authenticator Invalidation Manipulation

▼ PDU SecOC Freshness Overwrite Value Manipulation

Function Port	Substitute Value	Actual Value
Current Countdown Value		---
Countdown Start Value	5	---
Enable	0: Disabled	No conversion
Enable State		---
Freshness Overwrite Value	0	---

CanCluster - BusConfiguration_TX - IPdu02_FVCounter_secured (0x011) x +

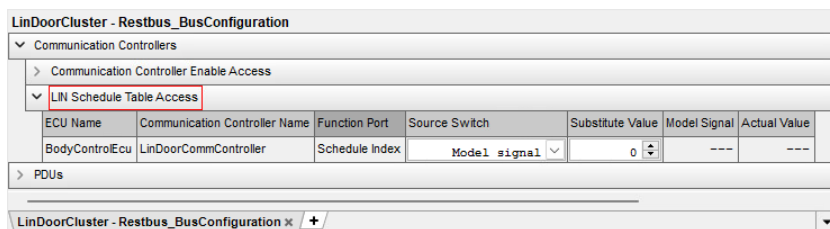
For more information, refer to [Manipulation Instrument for CAN \(Bus Manager\) \(ControlDesk Bus Navigator !\[\]\(8d0f0e0fe25b320c33272c52aec1fbca_img.jpg\)](#)).

LIN schedule table access support (Bus Manager applications only)

As of ControlDesk 2023-B, the TX Status instrument for LIN generated for bus communication modeled with the Bus Manager supports the *LIN schedule table*

feature by displaying the new LIN Schedule Table Access region that lets you change the active LIN schedule table during run time.

The following illustration shows the new instrument region.



For more information, refer to [TX Status Instrument for LIN \(Bus Manager\)](#) ([ControlDesk Bus Navigator](#)).

GTS communication manipulation support (dSPACE FlexRay Configuration Package-based applications only)

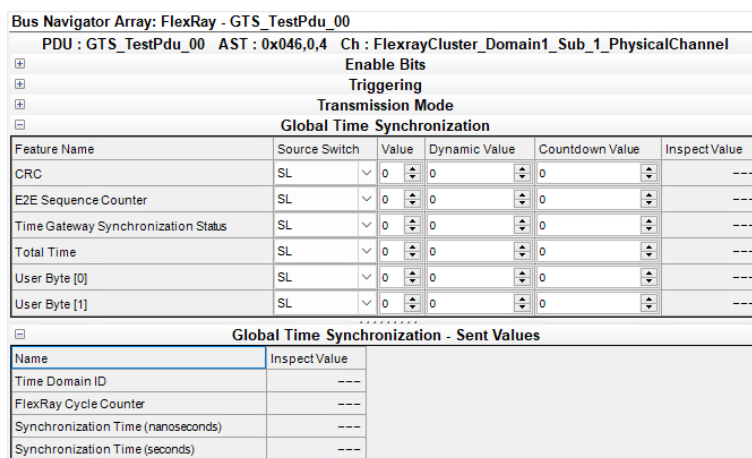
As of ControlDesk 2023-B, the TX and RX instruments generated for bus communication modeled with the dSPACE FlexRay Configuration Package support the *display and manipulation of global time synchronization (GTS)* via the new Global Time Synchronization and Global Time Synchronization - Sent Values instrument regions.

- TX instrument for FlexRay:
 - The Global Time Synchronization region lets you manipulate the values of time synchronization variables to be transmitted to the FlexRay bus.
 - The Global Time Synchronization - Sent Values region displays the values of time synchronization variables transmitted to the FlexRay bus.

RX instrument for FlexRay:

- The Global Time Synchronization region displays the values of time synchronization variables received from the FlexRay bus.

The following illustration shows the new Global Time Synchronization and Global Time Synchronization - Sent Values instrument regions of the TX Instrument for FlexRay as an example.



For more information, refer to:

- [TX Instrument for FlexRay \(Blockset\)](#) ([ControlDesk Bus Navigator](#))
- [RX Instrument for FlexRay \(Blockset\)](#) ([ControlDesk Bus Navigator](#))

Display of the PDU state in the region header (Bus Manager applications only)

As of ControlDesk 2023-B, the PDU State region header of the TX Instrument and the Compact Bus Instrument generated for bus communication modeled with the Bus Manager display the PDU state so that you no longer have to expand the region.

As an example, the following illustration shows the PDU State region header of a PDU for which transmission is enabled.



Visualizing signals for which 'ISignal Value' is enabled (Bus Manager applications only)

You can now visualize TX and RX signals (signals of a TX and RX PDU) for which the *ISignal Value* (refer to [Working with ISignal Values \(Bus Manager Stand-Alone\) Implementation Guide](#)) feature is enabled by dragging them from the Bus Navigator tree on a layout.

For an animated graphic, refer to dSPACE Help.

Support of R22-11 ARXML files

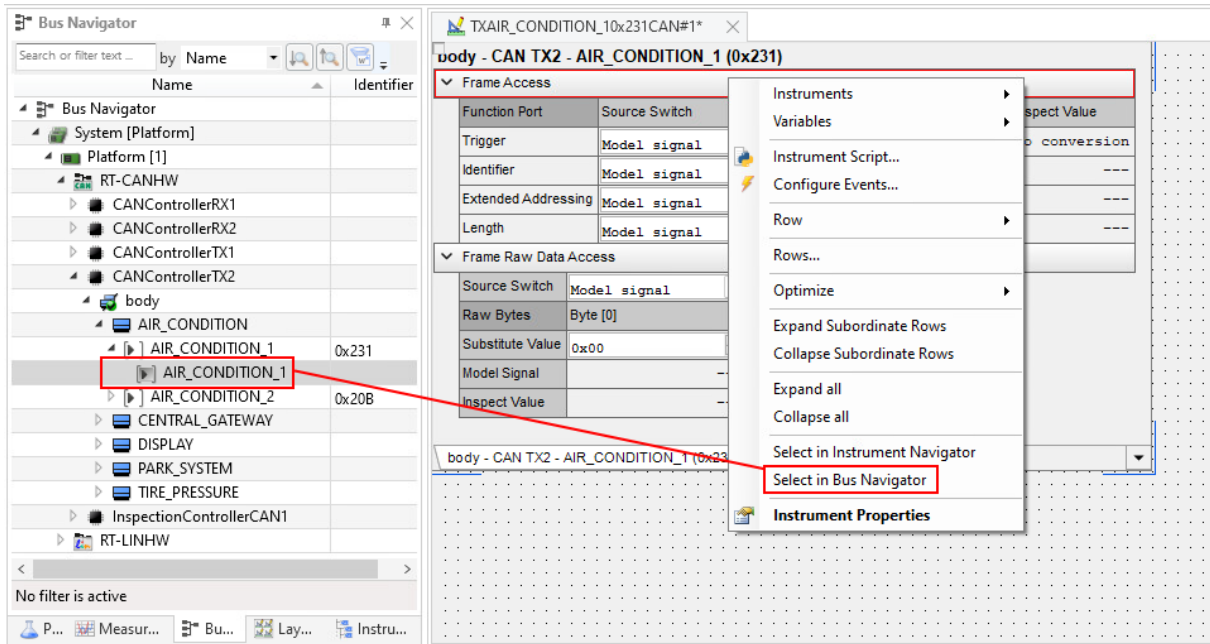
The ControlDesk Bus Navigator now also supports AUTOSAR system description (ARXML) files according to the AUTOSAR R22-11 Release for the CAN, LIN, and Ethernet Bus Monitoring devices.

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

Usability improvements

Selecting the Bus Navigator element related to a Bus instrument For bus communication modeled with the Bus Manager, the related Bus Instruments are based on the Hierarchy Array. You can now jump from such a Hierarchy Array to the related element in the tree of the Bus Navigator pane.

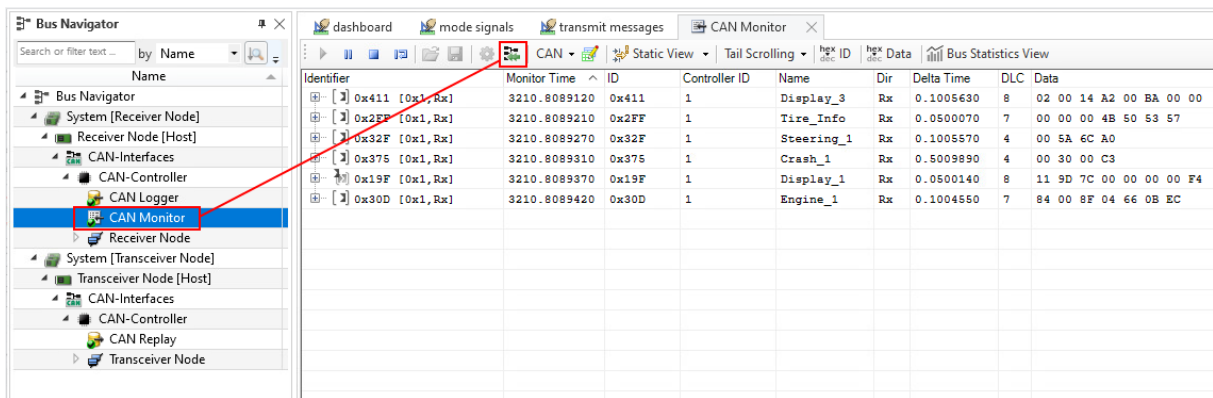
Refer to the following illustration. For an animated graphic, refer to dSPACE Help.



Refer to [Select in Bus Navigator \(ControlDesk Instrument Handling\)](#).

Selecting the monitor node related to a monitoring list You can now jump from a monitoring list to the related monitor node in the tree of the Bus Navigator pane.

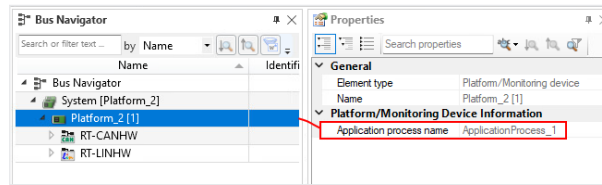
Refer to the following illustration. For an animated graphic, refer to dSPACE Help.



Refer to [Monitoring List \(ControlDesk Bus Navigator\)](#).

Display of the application process name for platforms When you select a platform element in the Bus Navigator tree, the ControlDesk Properties pane now displays the name of the related application process.

Refer to the following illustration for an example:



Compact Bus Instrument: Support for naming

macros ControlDesk 2023-B now supports naming macros also when you generate Compact Bus Instrument layouts.

You can specify naming macros on the Bus Navigator - Layout Generation page.

For instructions, refer to [How to Configure Layout Names \(ControlDesk Bus Navigator\)](#).

Related topics

References

- [Bus Navigator Page \(ControlDesk Bus Navigator\)](#)
- [Preview Features Page \(ControlDesk Customization\)](#)

New Electrical Error Simulation Features (ControlDesk 2023-B)

Support of the new SCALEXIO FSX hardware for fault simulation

ControlDesk 2023-B supports the new SCALEXIO FSX hardware for fault insertion, signal conditioning, and further extensions. The FSX hardware lets you use standard I/O boards for simulating faults in the wiring of an ECU.

The ControlDesk XIL API EESPort user interface lets you configure and switch these faults.

For more information on the new FSX hardware, refer to [Fault Simulation with FSX Boards \(SCALEXIO – Hardware and Software Overview\)](#).

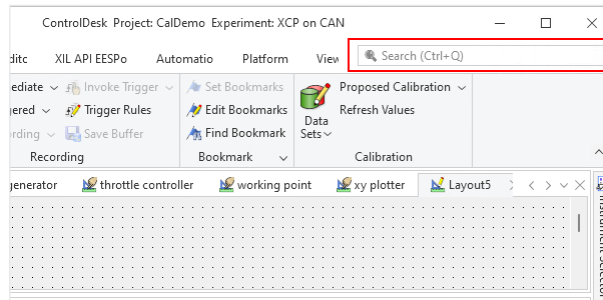
For more information on the ControlDesk XIL API EESPort user interface, refer to [Basics on Electrical Error Simulation \(ControlDesk Electrical Error Simulation via XIL API EESPort\)](#).

New User Interface Handling Features (ControlDesk 2023-B)

Searching for variables and block groups centrally

The ribbon in ControlDesk 2023-B provides a new field to search for variables and block groups in the Variables pane.

See the following illustration:



The names and paths of variables currently available in the Variables pane are searched.

In the search result list, you can do the following:

- Click a variable to select it in the variable list of the Variables pane.
- Click a block group to select it in the tree view of the Variables pane.
- Drag a variable to a layout to visualize it in an instrument.

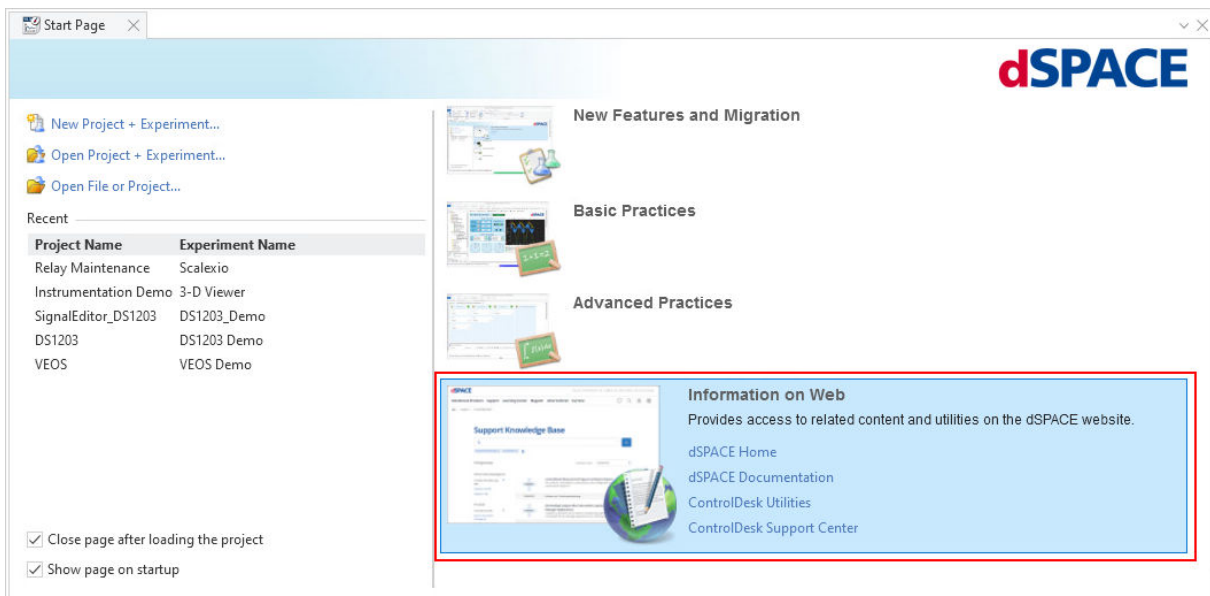
For an animated graphic, refer to dSPACE Help.

Refer to [Search \(ControlDesk User Interface Handling !\[\]\(ec9132f1d27c8919987d92907322654d_img.jpg\)](#)).

Access to content and utilities on the dSPACE website

The ControlDesk Start page now provides access to content and utilities on the dSPACE website.

Refer to the following illustration:



Refer to [Start Page \(ControlDesk User Interface Handling !\[\]\(fe3aebe81acea8d45108cd2768939da7_img.jpg\)](#)).

Preview features

As of ControlDesk 2023-B, ControlDesk provides *preview features*. Preview features are features in an early but completely tested development stage. They might benefit from user feedback before being finally integrated into the regular product.

You can enable preview features individually on the new [Preview Features page](#).

To provide feedback on or to request a feature, or report a problem, contact us by sending an e-mail to support@dspace.de.

Note

Due to the further feature development, there might be incompatible changes between a preview feature and the first stable version of that feature.

Related topics

References

[Preview Features Page \(ControlDesk Customization !\[\]\(8bba887393ca45b761e5cb49e755e762_img.jpg\)](#))

Migrating to ControlDesk 2023-B

Migrating to ControlDesk 2023-B

Introduction

To migrate from ControlDesk 2023-A to ControlDesk 2023-B and reuse existing experiments, you might have to carry out the following migration steps.

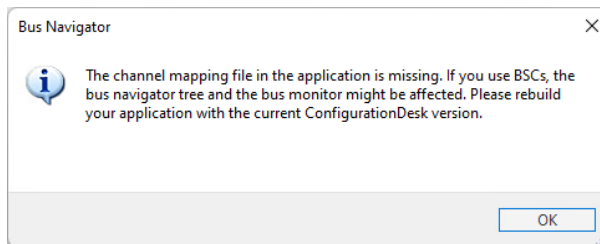
Note

To migrate to ControlDesk 2023-B from versions earlier than 2023-A, you also have to perform the migration steps of the intervening ControlDesk versions.

Rebuilding applications with bus configurations

As of dSPACE Release 2023-B, ConfigurationDesk and VEOS create the `EXPSWCFG.MAPPING` file during the build process of Bus Manager-based applications. ControlDesk 2023-B and later require this file to create a consistent Bus Navigator tree structure and to support bus monitoring for such applications.

If the `EXPSWCFG.MAPPING` file is missing, for example, if you load a bus configuration contained in an SDF file generated with dSPACE Release 2023-A or earlier, and if the bus configuration is based on a bus simulation container (BSC) file, ControlDesk cannot fill the Bus Navigator tree correctly, and you get the following message:



In this case, you have to rebuild the application with ConfigurationDesk or VEOS from dSPACE Release 2023-B or later.

Changed A2L file import

In ControlDesk 2023-B, the import of A2L files of version 1.6.1 or earlier containing elements with the `MATRIX_DIM` keyword was changed. This affects measurement arrays, value blocks, and struct arrays.

- Up to and including ControlDesk 2023-A, when you import such an A2L file, ControlDesk assigns a dimension to these elements even for a dimension value of 1 in the `MATRIX_DIM` keyword.
- As of ControlDesk 2023-B, when you import such an A2L file, ControlDesk ignores all dimension values of 1 in the `MATRIX_DIM` keyword. This behavior is compliant with the ASAM MCD-2 MC standard specification.

As a result, measurement arrays and value blocks can have fewer dimensions in ControlDesk 2023-B than in ControlDesk 2023-A or earlier. This affects variable accesses in automation scripts, ControlDesk layouts, etc.

Example The `Ca1Demo.a21` file of the `Ca1Demo` project contains the `MeasureArray` variable. The variable contains the following `MATRIX_DIM` keyword:

```
MATRIX_DIM 10 1 1
```

- The following illustration shows the `MeasureArray` variable in the Variables pane after A2L file import in ControlDesk 2023-A or earlier:

The following illustration shows the data type properties of the `MeasureArray` variable:

Data Type	
Data type	Array_UWORD_DIRECT_...
Byte order	MSB Last (Intel)
Range of values	[0, 65535]
Size in bytes	20
Dimensions	[10][1][1]

The following listing shows how to access a specific element of the `MeasureArray` variable with ControlDesk 2023-A or earlier:

```
...ActiveVariableDescription.Variables.ItemByPath('XCP()://MeasureArray[5][0][0]')
```

- The following illustration shows the `MeasureArray` variable in the Variables pane after A2L file import in ControlDesk 2023-B or later:

The following illustration shows the data type properties of the `MeasureArray` variable:

Data Type	
Data type	Array_UWORD_DIRECT_Row...
Byte order	MSB Last (Intel)
Range of values	[0, 65535]
Size in bytes	20
Dimensions	[10]

The following listing shows how to access a specific element of the `MeasureArray` variable with ControlDesk 2023-B or later:

```
...ActiveVariableDescription.Variables.ItemByPath('XCP()//MeasureArray[5]')
```

Tool automation changes

Migrating to Python 3.11 As of dSPACE Release 2023-B, dSPACE products no longer support Python 3.9. Instead, Python 3.11 is supported.

There are no ControlDesk-specific migration aspects.

For more information, refer to [Migrating Python Scripts from Python 3.9 to Python 3.11](#) on page 29.

Changed license handling and tool automation The license handling was changed in ControlDesk 2023-B.

As a result, if no valid license for the main version of ControlDesk is available, you can use only the following properties and methods of the `Application / IXaApplication <<Interface>>` interface:

- `CommandLineArguments`
- `Environment`
- `Execute(...)`
- `Log`
- `MainWindow`
- `Name`
- `Quit(...)`
- `UserSettings`
- `Version`
- `VersionInformation`

Change to Table Editor automation interfaces In ControlDesk 2023-B:

- The `Header` property was removed from the [TableEditorArrayInstrument / IViTableEditorArrayInstrument <<Interface>>](#) (ControlDesk Automation ) interface.
- The `InfoHeader` property was removed from the [Views / IViViews <<Interface>>](#) (ControlDesk Automation ) interface, which is returned by the [TableEditorInstrument / IViTableEditorInstrument <<Interface>>](#) (ControlDesk Automation ) interface.

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 New Features and Migration !\[\]\(e9474ce1d70442456f8fe9c393ea149c_img.jpg\)\)](#).

Related topics

Basics

[Basics on Migrating from Prior Versions of ControlDesk \(ControlDesk New Features and Migration !\[\]\(4146d17f71dced09c6ad789cacceaa6d_img.jpg\)\)](#)
[CalDemo \(ControlDesk Introduction and Overview !\[\]\(c0c268087214cb0c7c02d7259424fbe5_img.jpg\)\)](#)

References

[Application / IXaApplication <<Interface>> \(ControlDesk Automation 📖\)](#)
[Variables / IXaVariables <<Collection>> \(ControlDesk Automation 📖\)](#)

DCI Configuration Tool

New Features of the DCI Configuration Tool 2023-B

Firmware version for DCI-GSI2 interfaces

For the DCI-GSI2 interfaces, the firmware version 1.5.8 is delivered with the DCI Configuration Tool 2023-B.

Note

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

dSPACE ECU Flash Programming Tool

New Features of the dSPACE ECU Flash Programming Tool 2023-B

Use CAN MAX_DLC value

When configuring a flash project for which the XCP on CAN ECU interface has been selected, the dSPACE ECU Flash Programming Tool now lets you specify whether to use the specified maximum data length code (DLC) value or the actual used length for CAN/CAN FD messages.

Refer to [Configure Interface Settings Dialog \(ECU Flash Programming !\[\]\(74d4806277d7e73349d8e8c0897931e9_img.jpg\)](#)).

dSPACE FlexRay Configuration Package

New Features of dSPACE FlexRay Configuration Package 2023-B

FlexRay Configuration Package

Support of AUTOSAR Classic Platform R22-11 The FlexRay Configuration Package supports the format of AUTOSAR System Templates based on AUTOSAR Classic Platform Release R22-11 for describing FlexRay networks. However, no new features of AUTOSAR Classic Platform Release R22-11 are supported. Refer to [Communication Cluster Files Usable for Configuration \(FlexRay Configuration Tool Guide !\[\]\(9dfdaff1d86ba3c1f8353b4d1b61b8c5_img.jpg\)](#)).

FlexRay Configuration Tool

Enhancement to Create Corresponding ECU command Up to and including FlexRay Configuration Tool 2023-A, the execution of the Create Corresponding ECU command was possible only for ECUs that were not yet added to the Configuration view.

As of FlexRay Configuration Tool 2023-B, you can also have a restbus configuration created for ECUs that are already in the simulation configuration. If you have given the permission in the General Properties dialog, the FlexRay Configuration Tool lets you call the Create Corresponding ECU command also on ECUs that are already in the Configuration view.

Refer to [How to Configure a Restbus Simulation \(FlexRay Configuration Tool Guide !\[\]\(2b376d1a92330ab09dad2665d2f89bf5_img.jpg\)](#)).

dSPACE Installation Manager

Where to go from here

Information in this section

[New Features of dSPACE Installation Manager 23.2](#)..... 113

[Migrating to dSPACE Installation Manager 23.2](#)..... 114

New Features of dSPACE Installation Manager 23.2

Introduction

dSPACE Installation Manager 23.2 provides no new features that are relevant for users. However, note the following discontinuations that exist since version 23.1.

Discontinuation of the dongle migration with dSPACE Installation Manager

As of dSPACE Release 2023-A, dSPACE Installation Manager no longer supports migration from old dongles (delivered for dSPACE Release 2017-A and earlier) to currently delivered CmDongles.

If you still have an old dongle that needs to be migrated to use dSPACE Release 2017-B or later, the following applies:

- You are still using a WibuKey dongle (Type 1): Migration is no longer possible. A dongle exchange by dSPACE is required. For more information, refer to <http://www.dspace.com/go/DongleReplacement>.
- You are still using a first-generation CmDongle (Type 2), with serial number starting with 2-xxxx and white connector: Migration is no longer possible. A dongle exchange by dSPACE is required. For more information, refer to <http://www.dspace.com/go/DongleReplacement>.
- You are still using a CmDongle (Type 3), with serial number starting with 3-xxxx and old firmcode: Migration is required by using an earlier version of dSPACE Installation Manager (from 5.0 ... to 22.3). For the download of dSPACE Installation Manager 22.3, refer to <https://www.dspace.com/go/IM>.

Discontinuation of legacy licensing technologies

As of dSPACE Release 2023-A, dSPACE Installation Manager no longer supports legacy licensing technologies (WibuKey from Wibu-Systems for dongle licenses and FlexNet from Flexera for floating network licenses). This means:

- dSPACE Installation Manager 23.1 (on dSPACE Release 2023-A) and later will support only the CodeMeter licensing technology introduced with dSPACE Release 2017-B.
- dSPACE Installation Manager 23.1 and later no longer provides the file download (keys.dsp and license.dsp) for legacy licensing based on CodeMeter licenses.
- Installing dSPACE Release 2023-A or later on the same PC as dSPACE Release 2017-A or earlier is no longer supported and therefore not recommended.
- As of dSPACE Release 2023-A, dSPACE License Manager (Legacy) is no longer shipped and installed with the dSPACE Installation Manager. The associated command line utilities are also discontinued, including in particular the `IMLicUtil.exe` and the `InstallationReporter.exe` utilities.
- If you still need to use the legacy licensing method, you must use the latest version of dSPACE Installation Manager that supports this technology. This is dSPACE Installation Manager 22.3. You can download dSPACE Installation Manager 22.3 from <https://www.dspace.com/go/IM>.

Migrating to dSPACE Installation Manager 23.2

Using CmDongles

If you want to work with licenses on CmDongles in combination with dSPACE Installation Manager 23.2, for example, to activate, deactivate, or update licenses, the dongles must have at least 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 minimum firmware version.

dSPACE Installation Manager checks if the firmware of a connected dongle matches the required minimum 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 !\[\]\(6059a5aa8b4ca7bb793408023d6c6e42_img.jpg\)](#)).

Migration to Microsoft .NET6 framework

As of dSPACE Installation Manager 23.1, the dSPACE Installation Manager has been migrated to the Microsoft .NET6 framework. To achieve downward compatibility, dSPACE Installation Manager also supports the .NET4.8 framework until dSPACE Release 2026-B. Therefore, the API of the dSPACE Installation Manager can be used in .NET4.8 processes as well as in .NET6 processes.

Note on using the API of dSPACE Installation Manager: .NET4.8 (API version 2.0) will no longer be supported in a later release and you will then have to migrate the scripts and programs to API version 3.0. Therefore, for new scripts, already use API version 3.0.

dSPACE Python Extensions

New Features of dSPACE Python Extensions 2023-B

General enhancements

Support of Python 3.11 dSPACE Python Extensions supports Python 3.11 and the included pythonnet 3.0 package. For more information, refer to [Migrating Python Scripts from Python 3.9 to Python 3.11](#) on page 29.

dSPACE XIL API .NET

Where to go from here

Information in this section

[New Features of dSPACE XIL API .NET 2023-B](#)..... 117

[Migrating to dSPACE XIL API .NET 2023-B](#)..... 118

New Features of dSPACE XIL API .NET 2023-B

General enhancements

Support of Python 3.11 dSPACE XIL API .NET supports Python 3.11 and the included pythonnet 3.0 package. For information on the required migration steps, refer to [Product-Specific Information on Migrating](#) on page 33.

Enhanced MAPort functionality

Support of a new simulation platform The MAPort implementation of dSPACE XIL API .NET supports using the new MicroLabBox II as the simulation platform. For more information on the new MicroLabBox II, refer to [General Enhancements and Changes](#) on page 13.

For more information, refer to [dSPACE XIL API MAPort Implementation](#) .

Enhanced EESPort functionality

Support of new EES hardware The EESPort implementation of dSPACE XIL API .NET supports using the new SCALEXIO FSX hardware for fault simulation.

For more information, refer to [dSPACE XIL API EESPort Implementation](#) .

Enhanced ECUPort functionality

ECUPort Capture to MF4 You can now capture the data from a V-ECU and store it in an MF4 file on the host PC. The current implementation of storing to RAM or MF4 only supports the direct start and stop of a capture without triggering.

For more information, refer to [dSPACE XIL API ECUPort Implementation](#) .

dSPACE XIL API .NET on Linux

Discontinuation of the Python client The dSPACE XIL API .NET on Linux no longer contains the dSPACE-specific Python client. The Python access in Linux has been adapted to the access in Windows via pythonnet. For information on the required migration steps, refer to [Migrating to dSPACE XIL API .NET 2023-B](#) on page 118.

Migrating to dSPACE XIL API .NET 2023-B

Migration issues for dSPACE XIL API .NET on Linux

Due to the discontinued Python client, you must migrate your Python-based XIL API application to pythonnet. Using the pythonnet package, you must then adapt the handling of the return values, which were formerly Python objects and now .NET objects.

Using the pythonnet package The following example shows how to import the required clr module and how to load the ASAM assemblies:

```
import pythonnet
pythonnet.load("coreclr")
import clr

# Load ASAM assemblies
clr.AddReference("ASAM.XIL.Implementation.TestbenchFactory,
Version=2.2.0.0, Culture=neutral, PublicKeyToken=f9604847d8afbfb")
clr.AddReference("ASAM.XIL.Interfaces, Version=2.2.0.0,
Culture=neutral, PublicKeyToken=f9604847d8afbfb")
```

Using pythonnet and loading the assemblies in Linux is now similar to the use in Windows except for the following items:

- You must set the runtime by calling `pythonnet.load("coreclr")`.
- The ASAM does not provide assemblies for the Linux operating system. Therefore, you must reference the dSPACE-specific assemblies by specifying `f9604847d8afbfb` as the `PublicKeyToken` argument.

Handling return values Using pythonnet, the return values of XIL API methods and properties are .NET objects, such as `IList` and `IDictionary` instead of the Python data types `PyList` and `PyDict`.

Due to the changed types of return values, you must migrate your scripts, for example:

- Using Python data types in XIL API .NET up to dSPACE Release 2023-A

```
# Example for PyList
variableNames = CaptureResult.GetVariableNames(...)
numVariables = len(variableNames)
variableNames.append(newValue)
variableNames.sort()
# Example for PyDict
watcherDefines = Watcher.Defines
numDefines = len(watcherDefines)
```

- Using .NET objects in XIL API .NET as of dSPACE Release 2023-B

```
# Example for IList
variableNames = CaptureResult.GetVariableNames(...)
numValues = variableNames.Count
variableNames.Add(newValue)
variableNames.Sort()
# Example for IDictionary
watcherDefines = Watcher.Defines
numDefines = watcherDefines.Count
```

Here are some examples of XIL API methods and properties, which are affected.

Method/Property	Return Value Data Type	
	Up to Release 2023-A	As of Release 2023-B
Capture.Variables	PyList	IList<DataType>
Capture.Variables2	PyList	IList<DataType>
CaptureResult.GetVariableNames	PyList	IList<DataType>
MAPort.TaskInfos	PyList	IList<DataType>
VectorValue.Value	PyList	IList<DataType>
CaptureResult.MetaData	PyDict	IDictionary<DataType>
Watcher.Defines	PyDict	IDictionary<DataType>

ECU Interface Manager

Where to go from here

Information in this section

New Features of ECU Interface Manager 2023-B	121
An overview of the new features of ECU Interface Manager 2023-B.	
Compatibility of ECU Interface Manager 2023-B	122
Provides information on the compatibility of ECU Interface Manager 2023-B.	
Migrating to ECU Interface Manager 2023-B	122
Information on how to migrate to ECU Interface Manager 2023-B.	

New Features of ECU Interface Manager 2023-B

XCP time stamp synchronization

If an ECU application with integrated XCP service supports time-stamping, i.e., if the ECU application A2L file contains the `TIMESTAMP_SUPPORTED` entry, you can have the XCP time stamps synchronized with the system time of the dSPACE real-time hardware connected to the ECU. This also lets you time-correlate ECU data measured via XCP with data from other sources that are also connected to the dSPACE hardware.

Refer to [XCP Time-Stamping and Measuring ECU Data \(ECU Interface Manager Manual !\[\]\(17413706fd4997a1a4bdf85c6864eee1_img.jpg\)](#)).

Support of the new MicroLabBox II

The ECU Interface Manager lets you prepare external ECU interfacing using ConfigurationDesk and the new MicroLabBox II:

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

The MicroLabBox II supports the following ECU interfaces:

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

Compatibility of ECU Interface Manager 2023-B

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 ...			
	Version 2.11 ¹⁾	Version 2022-B ²⁾	Version 2023-A ³⁾	Version 2023-B ⁴⁾
ConfigurationDesk 2023-B ⁴⁾	✓ ⁵⁾	✓ ⁵⁾	✓ ⁵⁾	✓ ⁵⁾
ConfigurationDesk 2023-A ³⁾	✓ ⁵⁾	✓ ⁵⁾	✓ ⁵⁾	✓ ⁵⁾
ConfigurationDesk 2022-B ²⁾	✓ ⁵⁾	✓ ⁵⁾	✓ ⁵⁾	✓ ⁵⁾
ConfigurationDesk 22.1 ¹⁾	✓ ⁵⁾	✓ ⁵⁾	✓ ⁵⁾	✓ ⁵⁾

1) dSPACE Release 2022-A

2) dSPACE Release 2022-B

3) dSPACE Release 2023-A

4) dSPACE Release 2023-B

5) For 32-bit and 64-bit target architectures.

Compatibility between EIC files and RTMaps

RTMaps supports EIC files created with the ECU Interface Manager 2.10 and later.

Migrating to ECU Interface Manager 2023-B

Automatic migration of projects

You can reuse projects in the ECU Interface Manager 2023-B 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 2023-B, they are migrated automatically.

Note

In the ECU Interface Manager 2023-B, 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 2023-B from versions earlier than the ECU Interface Manager 2023-A, you might also have to perform the migration steps of the intervening ECU Interface Manager versions.

Firmware Manager

New Features of Firmware Manager 23.2

Changed default firmware archives for SCALEXIO and MicroAutoBox III

Up to and including dSPACE Release 2023-A, the preselected firmware archives in the Available archive versions dialog for SCALEXIO and MicroAutoBox III were the firmware archives of the active dSPACE Release. With dSPACE Release 2023-B, there is no longer a dependency to the active release, if it is later than Release 2021-B. The preselected firmware archive is then the latest archive version available in the installation folder.

The default SCALEXIO firmware archive now contains the 64-bit Linux-based firmware components. To load the 32-bit Linux-based firmware components, you have to explicitly select the 32-bit firmware archive.

Enhanced platform support

The Firmware Manager supports the update of the MicroLabBox II.

FPGA Programming Blockset

Where to go from here

Information in this section

[New Features of the FPGA Programming Blockset 2023-B..... 127](#)

[Migrating to the FPGA Programming Blockset 2023-B..... 129](#)

New Features of the FPGA Programming Blockset 2023-B

Extended FPGA design tool support

The FPGA Programming Blockset now supports the following products:

Design Tool Version	MATLAB Version ¹⁾	Operating System
AMD Vivado 2023.1 ²⁾	MATLAB R2022a	<p>Windows operating system that is supported by the RCP and HIL software of the current Release.</p> <p>For a list of supported operating systems, refer to Operating System on page 261:</p> <ul style="list-style-type: none"> ▪ The listed Windows Server operating systems are not officially supported by AMD, but tested by dSPACE. ▪ The AMD Vitis Model Composer does not officially support Windows 11, but it is tested by dSPACE.

¹⁾ The Processor Interface sublibrary of the FPGA Programming Blockset also supports MATLAB R2022b, R2023a, and R2023b.

²⁾ In general, Vivado ML Enterprise edition and the AMD Vitis Model Composer are needed. The Vivado ML Standard edition with AMD Vitis Model Composer can also be used for the DS2655 (7K160) and DS6601 FPGA base boards. The AMD Vitis Model Composer is required only for modeling FPGA applications with the FPGA Programming Blockset.

For more information on the software tools for FPGA programming, refer to [Software Tools for Working with the FPGA Programming Blockset \(FPGA Programming Blockset Guide !\[\]\(d263118e0bfd47dc6bc704167d936b83_img.jpg\)\)](#).

MATLAB compatibility The limited MATLAB compatibility reflects the requirements of the AMD Design Tools and its impacts on the building processes for creating an FPGA application.

In contrast to the FPGA build process, the blocksets and tools for modeling and building the processor application support the regular MATLAB compatibility of a dSPACE Release. Therefore, you can extend the MATLAB compatibility if you model and build the processor application separately from the FPGA application. The only requirement is that the same dSPACE Release is used to build the FPGA application and the processor application.

For example: The FPGA application is created with an installation set containing the current dSPACE Release, Vivado 2023.1, and MATLAB R2022a. You can transfer and import the generated FPGA application to an installation set containing the current dSPACE Release and MATLAB R2023b to continue your work with ConfigurationDesk or the Processor Interface sublibrary of the FPGA Programming Blockset.

General enhancements to the FPGA Programming Blockset

Incremental build The FPGA Programming Blockset now lets you enable the incremental build feature of Vivado to help in consistently achieving successful builds at high FPGA utilizations. A known good reference build is used as a starting point for implementation, so that new builds can assume a working FPGA configuration and only the actual changes need to be implemented. An incremental build increases the likelihood of successful FPGA builds and reduces build times for rather small changes in the FPGA model.

Refer to [How to Build FPGA Applications Incrementally \(FPGA Programming Blockset Guide !\[\]\(17acf1afa8cdf0b67c53d4865a5ed469_img.jpg\)\)](#).

dSPACE Timing Analyzer The new dSPACE Timing Analyzer helps you to fix timing errors that are detected during the build process.

The timing analyzer helps you with the following features:

- Shows the most critical FPGA signal paths of the built FPGA application independently of the used block libraries (HDL library of the AMD Vitis Model Composer and/or standard Simulink library (HDL Coder subsystems)).
The number of FPGA signal paths is limited to 1,000 bits of signals. For example, 50 20-bit wide data paths can be displayed.
- Provides details on the used blocks within a failed signal path.
- Opens a selected block of a failed signal path in the FPGA model. This gives you a fast access to fix the error.
- Provides the timing report with details on the selected signal path. The details help to fix errors that require detailed analysis.

For more information, refer to [Features of the dSPACE Timing Analyzer \(FPGA Programming Blockset Guide !\[\]\(d8ab143e904bfa3467271eec5af75a9b_img.jpg\)\)](#).


Message handling The FPGA Programming Blocks now provides a new message handling that informs you about possible problems or upcoming actions. The messages are issued in the MATLAB Command Window or in dialogs. They provide links and/or buttons for you to open the related topic in

the new FPGA Programming Blockset Message Reference, which contains more information.

New MicroLabBox II framework

The MicroLabBox II provides an FPGA on the DS1303 Multi-I/O Board that can be programmed with the FPGA Programming Blockset. The FPGA custom function block type built with the FPGA Programming Blockset replaces the standard I/O functionality of the I/O board. In ConfigurationDesk, you can either use the FPGA custom function block type or the function block types provided by ConfigurationDesk to implement the I/O functionality.

For an overview of the provided I/O functionality, refer to [Block Settings for the MicroLabBox II Framework \(FPGA Programming Blockset - FPGA Interface Reference\)](#) .

If you handcode the FPGA application, use the provided Tcl script to build the FPGA application. Refer to [How to Build the FPGA Application \(FPGA Programming Handcode Framework Guide\)](#) .

Note

The framework will be available after the installation of a service pack. The [service pack](#) will be released together with the MicroLabBox II.

Enhancements to the SCALEXIO and MicroAutoBox III frameworks

The frameworks now provide clock signals and additional information for subsystems using a user clock domain:

- Clock signal with the base rate of the FPGA.
- Clock signal of another subsystem using a user clock domain.
- System information on the FPGA application, the processor application, and an optional MGT module.


For more information, refer to [Using Clock Signals of Different Clock Domains in the Same Subsystem \(FPGA Programming Blockset Guide\)](#) .

Migrating to the FPGA Programming Blockset 2023-B

Migrating from previous FPGA Programming Blocksets

If you implemented an FPGA application with the FPGA Programming Blockset Version 1.1 or later and want to use it with the FPGA Programming Blockset 2023-B, the framework automatically updates to the current framework version.

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

You can also use a script to migrate processor interface blocks, for example, to migrate RTI processor models without the FPGA model. For more information, refer to [MigrateToModelPortBlocks \(FPGA Programming Software Script Interface Reference\)](#) .

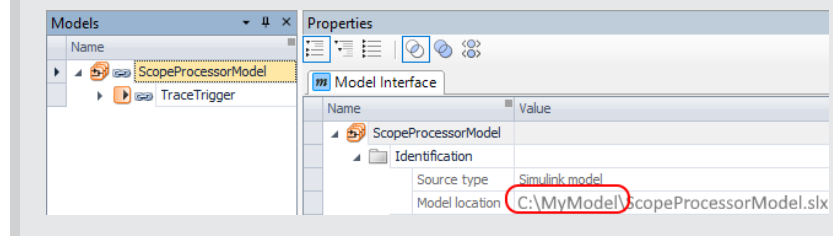
Changed process to generate FPGA scope rasters If you use an FPGA custom function built with the RTI FPGA Programming Blockset 3.13 (Release 2022-A) in ConfigurationDesk, you have to generate the FPGA scope raster that is used by ControlDesk.

As of FPGA Programming Blockset 2022-B, an FPGA custom function block with FPGA scope functions automatically generates the FPGA scope raster for ControlDesk.

If you use an existing ConfigurationDesk project and update the FPGA custom function of Release 2022-A with a newer FPGA custom function, the existing FPGA scope raster (<application>_usr.trc) causes an error.

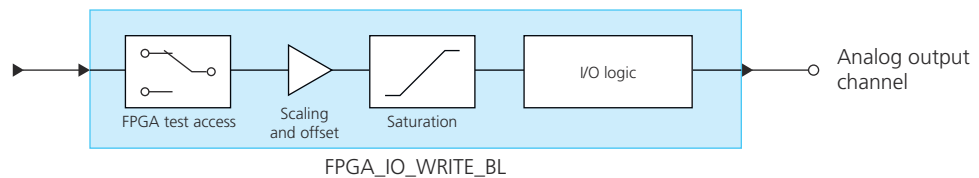
Note

Before you update the existing FPGA custom function in ConfigurationDesk with a new FPGAC file, delete the <application>_usr.trc file in the model folder. ConfigurationDesk shows the location of the model.



Changed interface to modify analog output signals FPGA scaling lets you modify analog output signals. As of Release 2022-B, the interface scales the signal values that are replaced with FPGA test access.

The following illustration shows the analog output interface. For more information on FPGA test access and scaling, refer to [Basics on FPGA Test Access and Scaling \(FPGA Programming Blockset Guide\)](#).



Up to Release 2022-A, the interface scales signal values first and then you can replace the scaled values with FPGA test access.

Bus In blocks replace Buffer64 blocks As of RTI FPGA Programming Blockset 3.12, the Bus In/Bus Out blocks for the MicroAutoBox III/SCALEXIO frameworks replace the transfer mode of the Buffer64 In/Buffer64 Out blocks. The update process automatically replaces the Buffer64 In/Buffer64 Out blocks that use the bus transfer mode with Bus In/Bus Out blocks.

Renaming of bus data transmission method As of FPGA Programming Blockset 23-A, the Synchronous to Read_Req method is named Synchronous to task method. The functionality does not change.

Rework necessary when migrating MGT In blocks Up to Release 2022-B, the MGT In block provides an Opto_Ready port. As of Release 2023-A, the

MGT In Opto Ready blocks provides the Opto_Ready port, because the signal of the Opto_Ready port cannot be provided in a user clock domain. After migration, you have to manually replace the MGT In block of the previous framework with a MGT In Opto Ready block and a MGT In block of the current framework.

Related topics

Basics

[Migrating and Updating Existing FPGA Models \(FPGA Programming Blockset Guide 📖\)](#)

MicroAutoBox III Firmware

New Features of the MicroAutoBox III Firmware 23.2

DS1521 Bus Board

The firmware of the DS1521 Bus Board now supports MACsec if the Ethernet controller supports this feature.

The type label on the bottom of the MicroAutoBox III shows whether the Ethernet controller supports MACsec via a footnote.

System Information

MicroAutoBox III		
Type	1403/1521 ¹⁾	1) MACsec
Serial	1007747	
Host-MAC	64:4D:70:00:E2:10	

Model Compare

Where to go from here

Information in this section

New Features of Model Compare 2023-B	135
Migration to Model Compare 2023-B	136

New Features of Model Compare 2023-B

Accelerated Model Comparison

Model Compare 2023-B provides a new, file-based comparison method that analyzes the pure model files directly and displays the comparison results immediately.

Related documentation

- [New Session: Select Method Dialog \(Model Compare Reference !\[\]\(2cbb40928a34ecf5ce700a63c52aa374_img.jpg\)](#))
- [New Session: Specify Model Files Dialog \(Model Compare Reference !\[\]\(ce05ba64c497267b6ad2e23c0c6ca4e1_img.jpg\)](#))
- [How to Create Comparison Sessions Using the Accelerated File-Based Comparison Method \(Model Compare Guide !\[\]\(3ce668af77d7d4613bd5f40b461563b3_img.jpg\)](#))

Textconv driver with built-in TargetLink support

As of Model Compare 2023-B, the Model2TextConverter (textconv driver) supports TargetLink block properties and parameters in detail.

Related documentation

- [Configure Model Compare as Textconv Driver \(Model Compare Guide !\[\]\(e84f4dc0518e2685b866048632d78d45_img.jpg\)](#))

Improved matching algorithm

With Model Compare 2023-B, the matching algorithm is further improved. Especially for model elements with few property changes and larger position changes, matching now works even better, so that fewer manual alignments are required.

Improved user documentation

With Model Compare 2023-B, an improved user documentation concerning workflow integration (e.g., using Git) of Model Compare is provided.

Related documentation

- [Using Model Compare in Common Workflows \(Model Compare Guide !\[\]\(48a7667d09d5a06397e047ee4537bb6f_img.jpg\)](#))

Migration to Model Compare 2023-B

Question

The `-textconv` option of `ModelCompare.exe` is replaced with the `DS.MOC.Model12TextConverter.CLI.exe`. It is recommended to use `ModelCompare.VCSTextConv.cmd`. If you already use it, no adaption is required.

Note

It is recommended to use only XML dump files that were created with this version of Model Compare. Otherwise, merge operations and some of the new features of Model Compare will not be available.

Model and Sensor Interface Blockset

Where to go from here

Information in this section

[New Features of the Model and Sensor Interface Blockset 2023-B..... 137](#)

[Migrating to Model and Sensor Interface Blockset 2023-B..... 137](#)

New Features of the Model and Sensor Interface Blockset 2023-B

Model and Sensor Interface Blockset

There are no new features in Model and Sensor Interface Blockset 2023-B.

Related topics

Basics

[Features of the Model and Sensor Interface Blockset \(Model and Sensor Interface Blockset Manual !\[\]\(d3102649f02e825ddb76dc3de0190154_img.jpg\)\)](#)

Migrating to Model and Sensor Interface Blockset 2023-B

MicroAutobox III

The MicroAutobox III platform is not supported by the Model and Sensor Interface Blockset.

For more information on using MicroAutobox III with the blockset, contact dSPACE Support.

Migrating from previous Releases

Migration from previous Releases as of dSPACE Release 2020-B Models created with previous Releases of the Model and Sensor Interface Blockset as of version 1.0 in dSPACE Release 2020-B are migrated automatically when you open the model. Messages are displayed in the MATLAB Command Window and in the migration log to confirm successful migration or notify you of failures.

Migration from the Model and Sensor Interface Blockset

Solution Models created with the Model and Sensor Interface Blockset Solution that was available prior to dSPACE Release 2020-B must be manually migrated. When you open a model that uses blocks from the blockset solution, a message is displayed with instructions on how to migrate the model to use the blocks of the latest Release of the Model and Sensor Interface Blockset.

ModelDesk

Where to go from here

Information in this section

New Features of ModelDesk 2023-B.....	139
Migration to ModelDesk 2023-B.....	140

New Features of ModelDesk 2023-B

General

The core technology of ModelDesk has been changed:

- It is modern to ensure future usability.
- It is updated for the application and parts of the user interface.
- It is prepared for modern operating systems and new .NET framework versions.
- It has unified user access where applicable.

Road Generator

OpenDRIVE files

- You can import roads in the OpenDRIVE format version 1.6.
- The quality of export of roads in OpenDRIVE format is improved.

Related topics

Basics

[Working with Roads in OpenDRIVE Format \(ModelDesk Road Creation !\[\]\(7d1d6890825e83a6a4a51febe2dcc7f3_img.jpg\)\)](#)

Migration to ModelDesk 2023-B

Project migration

As of ModelDesk 2023-B, you can migrate only projects created with ModelDesk version 5.2 (dSPACE Release 2019-B) and later.

Using an ASM of Release 2020-B in a migrated project

When you migrate a ModelDesk project and the corresponding ASM model, all signals in the layouts used for plotting are still connected. If you replace the migrated ASM model with an ASM model created with Release 2020-B, you must connect the signals again. ASM models created with Release 2020-B use `ASMSignalInterface` blocks that can be used in different parts of the model.

ASM project

Creating new ModelDesk projects on the basis of ASM projects is no longer supported for DS1006 target platforms. However, DS1006 platforms are still supported by ModelDesk. You can still parameterize real-time applications running on DS1006 platforms.

Model Interface Package for Simulink

Where to go from here

Information in this section

[New Features of the Model Interface Package for Simulink 2023-B.....](#) 141

[Migrating to the Model Interface Package for Simulink 2023-B.....](#) 142

New Features of the Model Interface Package for Simulink 2023-B

Enhanced support of Simulink root In and Out blocks

The Model Interface Package for Simulink assigns block and signal IDs to all external interfaces of the model. For Simulink® In, Out, Bus Element In and Bus Element Out ports, these IDs are now created using an improved schema. This results in the following enhancements:

- You can make changes in Simulink bus signals that are connected to these blocks without losing the connections in your ConfigurationDesk or VEOS projects.
- You can use identical names for inports and outports to easily identify the matching ports in case of model communication.

Note

- These enhancements are also effective for VEOS.
- For migration information, refer to [Migrating to the Model Interface Package for Simulink 2023-B](#) on page 142.

New MicroLabBox II platform support

MicroLabBox II and SCALEXIO have the same processor architecture and Linux64-based real-time operating system. Thus, Simulink models and SIC files generated for the SCALEXIO Linux 64-bit target architecture (platform identifier: SCALEXIO_LNX64) can now also be used in ConfigurationDesk applications for MicroLabBox II.

New VEOS_LINUX32_GCC platform identifier The Model Interface Package for Simulink now provides the new VEOS_LINUX32_GCC platform identifier for SIC files that are used in VEOS on Windows 32-bit.

Support of unbounded array variables The Model Interface Package for Simulink supports unbounded array variables in a Simulink model. However, you cannot use unbounded array variables in model port blocks and in Simulink root-level Inport or Outport blocks.

MATLAB compatibility The Model Interface Package for Simulink 2023-B is compatible with the following MATLAB® Releases:

- MATLAB R2023b
- MATLAB R2023a
- MATLAB R2022b
- MATLAB R2022a

Migrating to the Model Interface Package for Simulink 2023-B

Migrating to new ID behavior of Simulink root-level inports and outports The Model Interface Package for Simulink assigns new IDs to root-level Simulink® In, Out, Bus Element In, and Bus Element Out blocks. As a result, all connections to model ports in your ConfigurationDesk or VEOS projects, which relate to these Simulink blocks, will become unresolved after analyzing the model or updating the respective SIC file. You have to reconnect the model ports with the new IDs.

Note

You can restore the old ID schema to maintain the connections in your existing projects by setting the new UseClassicIdConventionForRootPorts parameter of the `dsmptb_pref()` API command in MATLAB to True:

```
dsmptb_pref('Set', 'UseClassicIdConventionForRootPorts', true);
```

Discontinuation of supported platforms The support of the SCALEXIO 32-bit Linux target platform (SCALEXIO_LNX) was removed with the Model Interface Package for Simulink 2023-B. The SCALEXIO 64-bit Linux target platform (SCALEXIO_LNX64) is still available.

MotionDesk

Where to go from here

Information in this section

[New Features of MotionDesk 2023-B.....](#) 143

[Migrating to MotionDesk 2023-B.....](#) 143

New Features of MotionDesk 2023-B

MotionDesk

There are no new features in MotionDesk 2023-B.

Migrating to MotionDesk 2023-B

Discontinuation of sensor simulation

As of MotionDesk Release 2023-A, sensor simulation based on MotionDesk is no longer supported for all sensor types.

AURELION dSPACE AURELION is the successor of MotionDesk for simulation animations and of Sensor Simulation for ADAS/AD simulations with camera, radar, and lidar sensors to validate driving functions and sensor output.

For more information and to prepare for your migration to AURELION, refer to <https://www.dspace.com/en/pub/home/support/kb/faqs/faq433.cfm> or contact dSPACE Support.

Planned discontinuations

MotionDesk MotionDesk and the MotionDesk Blockset will be discontinued after Release 2023-B. Until discontinuation, MotionDesk supports the simulation of camera sensors. Radar and lidar sensors are no longer supported using MotionDesk. For more information, refer to <https://www.dspace.com/go/elamd>.

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 endless sky in an old scene, you first have to delete these 3-D objects.

In advanced lighting mode, the static objects used for domes are not suitable for building the virtual world. You have to use the endless sky.

Migration of custom object VRML files

If you want to import 3-D custom objects in VRML2 format to MotionDesk, you must first convert the VRML2 files to the COLLADA (*.dae) format using MotionDesk up-to Release 2022-A. You can then import and use the COLLADA files in Release 2022-B and later.

You can convert the files in the Import Objects Dialog of the Library Manager Tool.

Migrating from previous Releases

MotionDesk supports experiments from the previous four versions of MotionDesk.

Real-Time Testing

Where to go from here

Information in this section

New Features of Real-Time Testing 2023-B..... 145

Migrating to Real-Time Testing 2023-B..... 145

New Features of Real-Time Testing 2023-B

New platform support

Real-Time Testing supports MicroLabBox II.

Migrating to Real-Time Testing 2023-B

Variables of the simulation application

The TRC file contains the variables of the simulation application that you can access via Real-Time Testing. Variables with rational scalings are no longer supported by Real-Time Testing.

Incompatible BCG files

BCG files generated on a Windows PC cannot be used on a Linux PC and vice versa. Generate the BCG files under the operating system that you want to use to manage real-time tests.

Only applicable for SCALEXIO as of Real-Time Testing 4.4 and VEOS as of Real-Time Testing 4.2: The internal Python interpreter version changed from 2.7.11 to 3.6.4. To use older scripts in the syntax of Python 2.7.11, you must migrate them to the syntax of Python 3.6.4. For more information on migrating Python scripts, refer to <http://www.dspace.com/go/Python36Migration>.

RTI/RTI-MP and RTLib

Where to go from here

Information in this section

[New Features of RTI/RTI-MP and RTLib.....](#) 147

[Migration Aspects of RTI/RTI-MP and RTLib.....](#) 147

New Features of RTI/RTI-MP and RTLib

New features of RTI/RTI-MP

RTI and RTI-MP has the following new feature:

- Support of MATLAB® R2023b.

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, 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.

RTI Bypass Blockset

Migrating to RTI Bypass Blockset 2023-B

Additional migration steps in some cases

To migrate to the RTI Bypass Blockset 2023-B from versions earlier than the RTI Bypass Blockset 2023-A and reuse existing models, you might have to carry out additional migration steps. For more migration information, refer to [History of Migration Steps \(RTI Bypass Blockset Reference !\[\]\(e2376d476d06eb31946dc01a69a4403a_img.jpg\)](#)).

Working with models from earlier RTI Bypass Blockset versions 2023-A, 2022-B, 3.x, and 2.x

The current Release contains RTI Bypass Blockset 2023-B, which is compatible with earlier blockset versions 2023-A, 2022-B, 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 from the prior RTI Bypass Blockset versions. If you have a Simulink model that was built with RTI Bypass Blockset 2.5 or earlier and you open it with RTI Bypass Blockset 2023-B, 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 when 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 2023-B 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 when 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 2023-A (.vdb) remains on the disk.

To enable the RTI Bypass Blockset to create the Data Dictionary again, 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 2023-A*

If a Simulink model was built with RTI Bypass Blockset 2.6 up to RTI Bypass Blockset 2023-A and you open it with RTI Bypass Blockset 2023-B, 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 and including RTI Bypass Blockset 2023-A, 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 2023-B.....](#) 151

[Migrating to RTI CAN MultiMessage Blockset 2023-B.....](#) 151

New Features of the RTI CAN MultiMessage Blockset 2023-B

Support of AUTOSAR Classic Platform R22-11

The RTI CAN MultiMessage Blockset supports the AUTOSAR System Template based on AUTOSAR Classic Platform Release R22-11 for describing CAN networks. However, no new features of AUTOSAR Classic Platform Release R22-11 are supported.

Refer to [General Settings Page \(RTICANMM MainBlock\)](#) (RTI CAN MultiMessage Blockset Reference .

Migrating to RTI CAN MultiMessage Blockset 2023-B

Planned discontinuation

RTI CAN MultiMessage Blockset for SCALEXIO is planned for discontinuation after Release 2024-B. For more information, refer to <https://www.dspace.com/go/elarTixxMMBS>.

RTI CAN MultiMessage Blockset should no longer be used in new SCALEXIO projects. For new projects, you are recommended to use the Bus Manager, which combines a multitude of functionalities in one tool.

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 \(RTI CAN MultiMessage Blockset Reference !\[\]\(750841ae7100dc832cb0a4b3af4492f3_img.jpg\)\)](#) block.

For more information, refer to [Limitations with RTICANMM \(RTI CAN MultiMessage Blockset Reference !\[\]\(17acf1afa8cdf0b67c53d4865a5ed469_img.jpg\)\)](#).

Additional migration steps in some cases

To migrate to the RTI CAN MultiMessage Blockset 2023-B from versions earlier than the RTI CAN MultiMessage Blockset 2023-A and reuse existing models, you might have to carry out additional migration steps. For more migration information, refer to [History of Migration Steps \(RTI CAN MultiMessage Blockset Reference !\[\]\(f95dab70c751fda7d824b8b03650f7aa_img.jpg\)\)](#).

RTI LIN MultiMessage Blockset

Where to go from here

Information in this section

[New Features of the RTI LIN MultiMessage Blockset 2023-B](#)..... 153

[Migrating to RTI LIN MultiMessage Blockset 2023-B](#)..... 153

New Features of the RTI LIN MultiMessage Blockset 2023-B

Support of AUTOSAR Classic Platform R22-11

The RTI LIN MultiMessage Blockset supports the AUTOSAR System Template based on AUTOSAR Classic Platform Release R22-11 for describing LIN networks. However, no new features of AUTOSAR Classic Platform Release R22-11 are supported.

Refer to [General Settings Page \(RTILINMM MainSetup\)](#) (RTI LIN MultiMessage Blockset Reference )

Migrating to RTI LIN MultiMessage Blockset 2023-B

Planned discontinuation

RTI LIN MultiMessage Blockset for SCALEXIO is planned for discontinuation after Release 2024-B. For more information, refer to <https://www.dspace.com/go/elarTixxMMBS>.

RTI LIN MultiMessage Blockset should no longer be used in new SCALEXIO projects. For new projects, you are recommended to use the Bus Manager, which combines a multitude of functionalities in one tool.

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 \(RTI LIN MultiMessage Blockset Reference !\[\]\(cf5be311f7b2821912d8009884508fa2_img.jpg\)\)](#) block.

For more information, refer to [Limitations of RTI LIN MultiMessage Blockset \(RTI LIN MultiMessage Blockset Reference !\[\]\(339a16584d5da0f0a3ca4e9ec17bf6a1_img.jpg\)\)](#).

RTI Synchronized Time Base Manager Blockset

New Features of the RTI Synchronized Time Base Manager Blockset 2023-B

Support of MicroLabBox II

The RTI Synchronized Time Base Manager Blockset supports the new MicroLabBox II.

SCALEXIO Firmware

Where to go from here

Information in this section

New Features of the SCALEXIO Firmware 23.2.....	157
Migrating to SCALEXIO Firmware 23.2.....	158
Discontinuations in SCALEXIO Firmware.....	159

New Features of the SCALEXIO Firmware 23.2

New supported hardware

The SCALEXIO firmware supports the following new hardware:

- DS6242 D/A Board

The DS6242 D/A Board is a single-slot SCALEXIO I/O board. It provides 10 galvanic-isolated analog channels for voltage out (1 channel up to 20 V, 3 channels in series up to 60 V) and current sink (1 channel up to 40 mA, 10 channels in parallel up to 400 mA).

- DS6365-PE Board

The DS6365-PE Automotive Ethernet Board is a PCIe 3.0 x4 board that provides two independent ports for connecting external devices to a SCALEXIO Processing Unit via an Automotive Ethernet connection with up to 10 Gbit/s.

- SCALEXIO FSX Hardware

For SCALEXIO a new product line is developed, called SCALEXIO FSX. The FSX abbreviation stands for fault insertion, signal condition, and further extensions. SCALEXIO standard I/O boards cannot be used for simulating faults in the wiring of an ECU. Until now, you could only do a fault simulation with a SCALEXIO system with HighFlex boards (e.g., a DS2601) or MultiCompact unit (e.g., a DS2680). When extending standard I/O boards with FSX boards, it is also possible to do fault simulation with standard I/O boards. In the first version, the FSX boards can only be used in customized SCALEXIO systems.

Together with a customized SCALEXIO system, you get a signal list. The signal list allows to configure the fault classes that are allowed to be switched.

For editing the signal list, the Signal List Editor is developed. The faults are switched via the XIL API EESPort. This is used by ControlDesk's Failure Simulation Package and can be used by automation tools.

Fault simulation with FSX boards is not compatible with fault simulation with HighFlex and MultiCompact hardware of SCALEXIO.

For details, refer to [Fault Simulation with FSX Boards \(SCALEXIO – Hardware and Software Overview\)](#)

- DS6511 Connector Panel

The SCALEXIO Connector Panel offers easy and convenient access to the I/O channels of standard I/O boards via BNC connectors. The connector panel is connected to the SCALEXIO I/O boards (DS6221, DS6241) by means of a 50-pin Sub-D ribbon cable.

Related topics

Basics

[DS6242 D/A Board \(SCALEXIO Hardware Installation and Configuration\)](#)
[DS6365-PE Automotive Ethernet Board \(SCALEXIO Hardware Installation and Configuration\)](#)

Migrating to SCALEXIO Firmware 23.2

Migrating to the 64-bit Linux-based operating system

With dSPACE Release 2022-A, the default SCALEXIO firmware is a 64-bit Linux-based distribution. The following items built for dSPACE Release 2021-B and earlier are no longer compatible with the SCALEXIO system and must be (re-)built from source code based on dSPACE Release 2022-A or later:

- Real-time applications
- Binary libraries contained in model containers (i.e., SIC, BSC, FMU, and CTLGZ files)
- SIC and BSC files
- Binary libraries referenced by Simulink models
- Binary libraries referenced by ConfigurationDesk applications via custom code settings or custom I/O functions

Source code is expected to be reusable in most cases. Cases that might require an adaptation of C code are related to custom code that uses special features (e.g., OS-specific functions) created by your company or third-party suppliers.

Hypervisor Extension

If the new SCALEXIO Hypervisor Extension is installed, SCALEXIO real-time applications as well as Linux real-time and non-real-time applications created for dSPACE Release 2020-A and earlier can no longer be used. You must rebuild the SCALEXIO real-time applications and adapt Linux real-time and non-real-time applications to the new hypervisor, especially regarding interrupts, shared memory, and the assignment of hardware resources.

Discontinuations in SCALEXIO Firmware

Processing Unit

The SCALEXIO Processing Unit with the Intel® Core™ i7-860 Processor (Real-Time PC Version 1.0) is not supported by the SCALEXIO firmware as of version 6.0 (dSPACE Release 2022-A).

Discontinuation of 32-bit support

The 64-bit Linux-based real-time operating system (RTOS) is the default RTOS for a SCALEXIO system and the DS6001 Processor Board. The 32-bit Linux-based RTOS is not supported.

Signal List Editor

Introduction

The Signal List Editor is a new dSPACE software product to be used in a fault simulation scenario with a SCALEXIO system.

Fault simulation means inserting electrical faults into the wiring of an ECU to analyze the behavior of the system when wiring problems occur. For example, you can simulate situations in which an ECU pin is short-circuited to ground or the battery voltage, or an ECU pin is not connected (broken wire, open circuit).

The Signal List Editor is used in such a scenario to specify the allowed fault classes for each signal, which can be used in the experiment software (ControlDesk or dSPACE XIL API) for simulating faults.

Note

The Signal List Editor supports only SCALEXIO FSX boards (for example, DS6801 Fault Insertion Board, DS6805 Fault Insertion Board). The HighFlex and MultiCompact SCALEXIO hardware that provides fault simulation (for example, the DS2680 I/O Unit) is not supported by the Signal List Editor. For this hardware, you have to use ConfigurationDesk instead.

Features of Signal List Editor 2023-B (23.1)

Main features

The main features of the Signal List Editor are:

- Displaying a signal list with configuration data required for electrical fault simulation.
- Extending this signal list by adding new external devices and new signals.
- Specifying the allowed fault classes for ECU signals to be simulated in the experiment software such as ControlDesk.
- Specifying the load rejection behavior for connected loads in case of fault simulation.

For more information, refer to [Introduction to the Signal List Editor \(Signal List Editor Manual !\[\]\(529949c2c3dadbaa4e538e8c643454bc_img.jpg\)](#)).

SYNECT

Migrating to SYNECT 2023-B

Where to go from here

Information in this section

Migrating from SYNECT 2023-A	164
Overview of steps required for migrating from SYNECT 2023-A.	
Migrating Databases	164
To use the data from previous SYNECT versions with SYNECT 2023-B, you have to migrate the SYNECT database.	
Data Model Changes from SYNECT 2023-A to SYNECT 2023-B	165
The data model did not change from SYNECT 2023-A to SYNECT 2023-B.	

Migrating from SYNECT 2023-A

Python 3.11 support

SYNECT now supports Python 3.11 that is distributed with dSPACE Release 2023-B. This can lead to migration issues for scripts using pythonnet. Refer to SYNECT-specific migration issues (refer to [Using SYNECT](#) on page 36).

Migrating Databases

Introduction

To use the data from previous SYNECT versions with SYNECT 2023-B, you have to migrate the SYNECT database.

To migrate databases from SYNECT Versions 2.0 - 2023-A to SYNECT 2023-B, use the Database Migrator of SYNECT 2023-B.

Note

Contact dSPACE Support if you want to migrate SYNECT versions earlier than SYNECT 2.0. Refer to www.dspace.com/go/supportrequest.

For basic information and instructions on migrating databases, refer to [Migrating Databases from Previous SYNECT Versions \(The SYNECT Server Guide !\[\]\(a8f9309f944226d1420f5fed22e2b6e6_img.jpg\)](#)).

Data Model Changes from SYNECT 2023-A to SYNECT 2023-B

Introduction

The data model did not change from SYNECT 2023-A to SYNECT 2023-B.

SystemDesk

Where to go from here

Information in this section

New Features of SystemDesk 2023-B.....	168
Migrating to SystemDesk 2023-B.....	172

New Features of SystemDesk 2023-B

Where to go from here

Information in this section

New General Features.....	168
New V-ECU Generation Features.....	169
Improved A2L File Handling.....	169

New General Features

Classic Platform support by SystemDesk 2023-B

AUTOSAR release for modeling SystemDesk lets you model Classic Platform software and system architectures with a data model according to the AUTOSAR R22-11 Release. However, SystemDesk also lets you exchange data of other AUTOSAR releases.

Data exchange support SystemDesk supports AUTOSAR R22-11, R21-11, R20-11, R19-11, 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 R22-11 for developing Adaptive Platform software. For exchanging data, AUTOSAR R22-11, R21-11, R20-11, R19-11, and R19-03 are supported.

Python support

dSPACE Release 2023-B now provides Python 3.11, which you can use for developing Python scripts to automate SystemDesk via its COM API.

The SystemDesk demo scripts did not need to be explicitly adapted for Python 3.11. This is also expected for custom automation scripts.

You have to migrate Python scripts of previous SystemDesk versions due to API changes. Refer to [Migrating API Scripts](#) on page 174.

New V-ECU Generation Features

Compatibility of V-ECUs generated with SystemDesk 2023-B

SystemDesk lets you export V-ECUs to be built and simulated in the following formats:

- *V-ECU implementation containers*: Version 3.3

V-ECU implementation containers are a dSPACE-specific format that supports all features of the dSPACE SIL solution. You can build and simulate version 3.3 V-ECUs with dSPACE VEOS 2023-B.

For detailed compatibility information, refer to [Compatibility of VEOS 2023-B](#) on page 253.

- *V-ECU FMUs*: FMI 2.0/3.0

You can export V-ECU FMUs with SystemDesk to support OEM-supplier scenarios where not all partners use the dSPACE SIL solution. You can build and simulate V-ECU FMUs with FMI-compliant simulators.

For detailed and up-to-date compatibility information on dSPACE FMI support, refer to <http://www.dspace.com/go/FMI-Compatibility>.

IronPython support

SystemDesk 2023-B now supports IronPython 3.4 for BSW module scripts, Custom module conversion scripts, and custom validation rules. However, in most cases, migration of previous SystemDesk versions IronPython 2.7 scripts is not required.

fmi-ls-xcp layered standard support

SystemDesk now supports the generation of V-ECU FMUs with an internal XCP service implementation according to the `fmi-ls-xcp` layered standard version 1.0.0-RC1. Refer to <https://github.com/modelica/fmi-ls-xcp>.

Improved A2L File Handling

Introduction

With this version of SystemDesk, handling A2L files was improved to better support AUTOSAR workflows for exchanging SWCs, the AUTOSAR methodology for generating A2L files, and overcome limitations regarding SWC-internal Measurement and Calibration (MC) variables.

- When generating A2L files for V-ECUs, SystemDesk now supports static and constant memories including curves and maps according to AUTOSAR. This allows you to define SWC-internal variables entirely on AUTOSAR arxml files. Exchange of SWC descriptions between SystemDesk and other AUTOSAR tools is now feasible without additional A2L files. Refer to [Static and constant memory support for A2L file generation](#) on page 170.
- SystemDesk continues the support for the integration of SWCs whose SWC-internal variables are described by A2L files. The import of A2L files now creates static and constant memories according to AUTOSAR. You can edit imported SWC-internal variables in SystemDesk and hand-over SWCs without

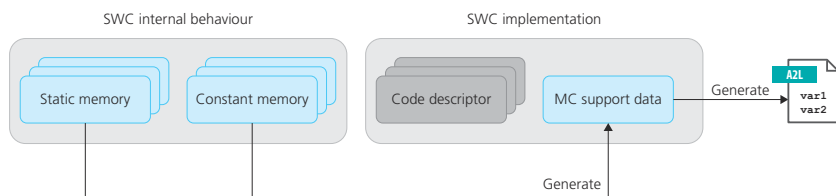
additional A2L files. Refer to [A2L file support for SWC integration](#) on page 170.

- SystemDesk continues the support for merging A2L files. You can reference existing A2L files at SWC implementations and directly at V-ECUs. These files are merged with the A2L fragments generated by SystemDesk. Refer to [Exporting V-ECU A2L files](#) on page 171.

Static and constant memory support for A2L file generation

SystemDesk now supports static and constant memories according to AUTOSAR when generating A2L files.

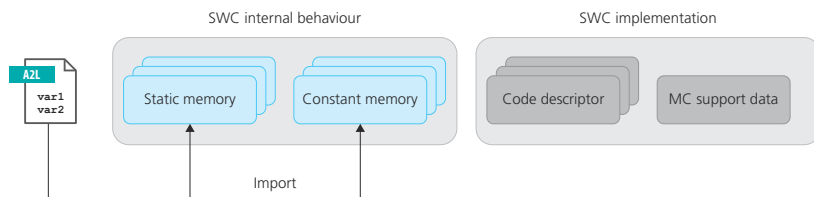
Static and constant memories let you use AUTOSAR data types in the implementation of SWCs without using RTE features. This enables MC support for SWC-internal variables.



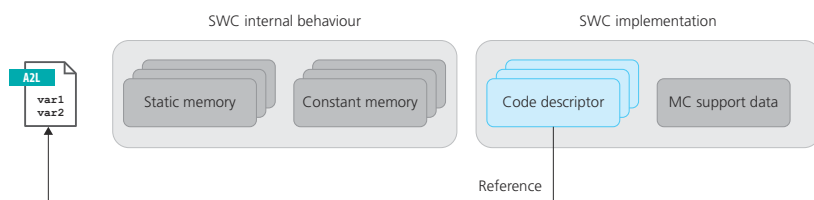
Generating A2L variable descriptions from static and constant memories is a two-step process. First, the MC support data, which is part of the SWC implementation, is generated from the static and constant memories in the AUTOSAR model. Then the A2L variable description is generated in a second step from the MC support data. SystemDesk provides generators for both steps, which are automatically used during V-ECU export.

A2L file support for SWC integration

SystemDesk continues the support for importing A2L files with SWC-internal variables when integrating SWCs into an AUTOSAR model. The import now creates static and constant memories according to AUTOSAR. This lets you edit measurements and characteristics with SystemDesk as required.



If editing SWC-internal variables is not your intention, you can reference A2L files as an alternative. You have to create code descriptors at the SWC implementation in this case. Refer to the following illustration.



Exporting V-ECU A2L files

You can use MC tools to access ECU variables during VEOS simulation. When exporting V-ECU implementation containers, SystemDesk merges all relevant variable descriptions.

Model-based V-ECUs You can use model-based V-ECUs in situations where you want to test application software. You can define static and constant memories in the AUTOSAR model, import A2L files to application layer SWCs, or reference the A2L files at SWC implementations. The generators for the RTE and the basic software components that dSPACE provides for SIL testing add A2L fragments.

SystemDesk merges the following variable descriptions:

- A2L fragments that SystemDesk generates based on MC support data for constant and static memories. SystemDesk generates the MC support data in a first step.
- A2L files you reference at SWC implementations.
- A2L fragments that SystemDesk generates based on MC support data for the RTE and basic software.
- A2L files you reference directly at the V-ECU.

Code-based V-ECUs You can use code-based V-ECUs in situations when the ECU integration has been done with the production BSW. You add relevant software packages including the A2L files to the V-ECU. dSPACE provides module configurations for the microcontroller abstraction layer (MCAL) that let you generate code and A2L fragments.

SystemDesk merges the following variable descriptions:

- A2L files of software packages.
- A2L fragments that SystemDesk generates for MCAL modules.

Migration

The improvements for A2L file handling replace a custom database for variables that is no longer part of SystemDesk.

Migration has the following aspects:

- Migrating projects is performed automatically when you open projects with the new version of SystemDesk. Imported A2L files are exported and referenced. Refer to [Migration of projects with imported A2L files](#) on page 173.
- You have to migrate API scripts that access SWC-internal variables. Refer to [Handling internal variables of SWC-internal behaviors](#) on page 177.

Migrating to SystemDesk 2023-B

Where to go from here

Information in this section

Discontinuations of dSPACE Release 2023-B.....	172
Migrating to SystemDesk 2023-B.....	173
Migrating API Scripts.....	174

Discontinuations of dSPACE Release 2023-B

Discontinued partial

AUTOSAR Import and Export commands

The following partial AUTOSAR Import and Export commands for software components are no longer available in SystemDesk 2023-B:

- Import – AUTOSAR, Update Component Ports
- Import – AUTOSAR, Update Internal Behavior
- Export – AUTOSAR, Ports Only
- Export – AUTOSAR, Internal Behavior Only

SystemDesk provides support for splittables that you can use instead. You have to assign AUTOSAR elements to master files and split software components and internal behaviors to different files as required.

No debugging of IronPython scripts

SystemDesk supports IronPython for the following use cases:

- *BSW module script*: To configure RTE and BSW modules according to the application software via context menu commands and the **Auto Configure** and **Generate** command.
- *Custom conversion script*: To convert module configurations for import/export scenarios without a loss of data.
- *Custom validation rule*: To specify custom rules for validation of SystemDesk project elements.

With previous SystemDesk versions, you were able to debug scripts with Microsoft Visual Studio and add breakpoints to scripts and debug them as required. However, with SystemDesk 2023-B debugging of IronPython scripts is no longer supported.

Discontinuation of views on ECU configurations

The ECU Configuration Manager no longer supports different views.

You cannot switch to the AUTOSAR View or Dependency View. However, all SystemDesk features for configuring ECUs are available in the default Flat List View.

Migrating to SystemDesk 2023-B

Automatic migration of projects

SystemDesk 2023-B automatically migrates SystemDesk 5.6, and 2022-B SDP project files when it starts.

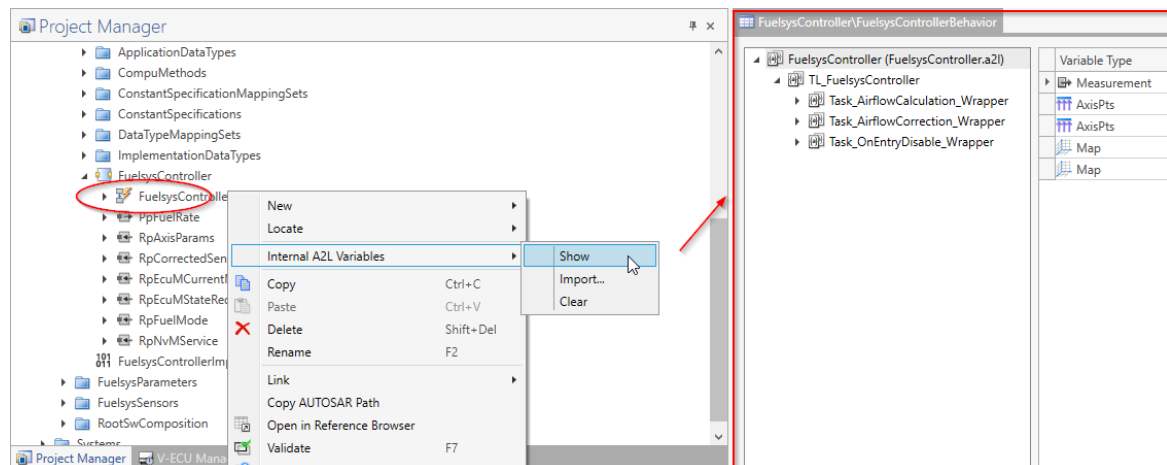
Note

You are recommended to install the most recent patch for SystemDesk 5.6 or 2022-B. Then, save the SDP project files you want to migrate before opening them in SystemDesk 2023-B.

Migration of projects with imported A2L files

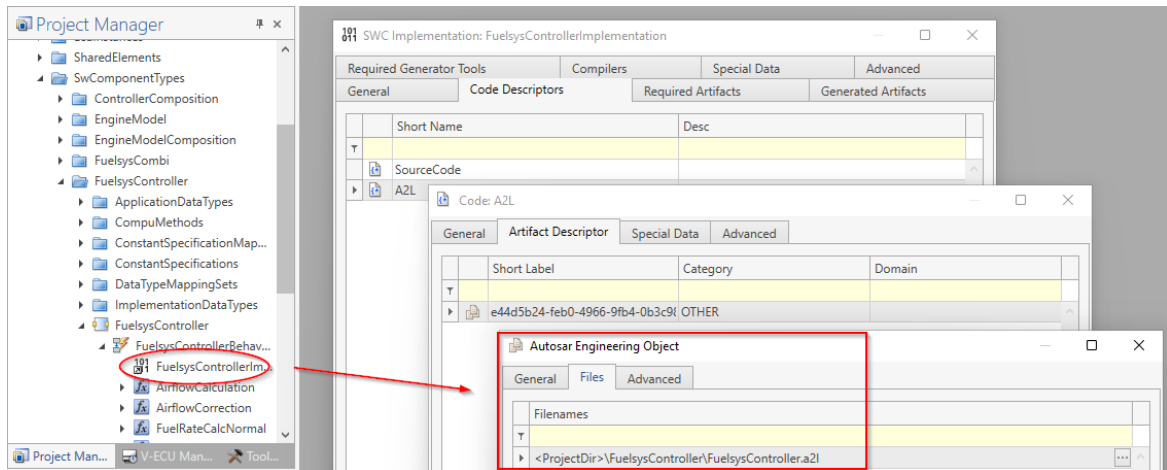
Handling A2L files has been changed with this version of SystemDesk. This has the following effect on migrated SystemDesk projects where one or more A2L files were imported at SWC-internal behaviors.

Old projects Refer to the following illustration showing internal variables at the SWC-internal behavior in the AR Fuelsys demo in SystemDesk 2022-B.



Migrated projects SystemDesk 2023-B automatically migrates SystemDesk 5.6 and 2022-B SDP project files. Internal variables of SWC-internal behaviors are automatically exported to A2L files in a subfolder of the folder that contains the SystemDesk project file. These A2L files are linked to the respective SWC implementation. This also applies to A2L files imported to V-ECUs that are now referenced in SystemDesk 2023-B.

Refer to the following illustration showing the migrated AR Fuelsys demo project in SystemDesk 2023-B.



Migrating API Scripts

Handling string properties

You have to adapt handling string properties. You cannot use the Python `None` keyword any longer. You can use the `pythoncom.Nothing` property instead.

Old script The following listing shows using the `None` keyword.

```
activeProject = systemDeskApplication.ActiveProject
root = activeProject.RootAutosar
packages = root.ArPackages
for package in packages.Elements:
    package.SdDescription = None
```

New script The following listing shows importing the `pythoncom` module and using the `pythoncom.Nothing` property.

```
import pythoncom

activeProject = systemDeskApplication.ActiveProject
root = activeProject.RootAutosar
packages = root.ArPackages
for package in packages.Elements:
    package.SdDescription = pythoncom.Nothing
```

Handling exceptions

- You have to adapt handling exceptions when using C# because the COM automation now returns only limited information on exception types. Therefore, you cannot respond to specific exception types, such as `ValueNullException` or `AutomationLockedException`. However, the exception messages allow you to respond to specific exceptions.
- Handling exceptions when using Python did not change because specific exception types were never provided and you always had to use exception messages for responding to specific exceptions.

Old listing The following listing shows handling exception using the exception type.

```
try
{
    bool demo = runnable.CanBeInvokedConcurrently;
}
catch (ValueNullException e)
{
    // do something
}
```

New listing SystemDesk returns `System.Exception` exceptions for all COM API exceptions. The following listing shows exception handling using the exception message.

```
try
{
    bool demo = runnable.CanBeInvokedConcurrently;
}
catch (System.Exception e)
{
    if (e.Message.Equals("The value is not set. ")) {
        // do something
    }
}
```

Migrating scripts for automating SystemDesk

The SystemDesk API was changed as of SystemDesk 2023-B. Some interfaces were added compared to SystemDesk 2022-B and certain interfaces were changed.

For more information, refer to *API Changes from SystemDesk 2022-B to SystemDesk 2023-B* in the *SystemDesk API Reference*, which is available only in dSPACE Help.

Changed type of the `IPortInterface.IsService` property One of the SystemDesk API changes introduced by the AUTOSAR R22-11 support concerns the `IsService` property of the `IPortInterface` of the software component template. You have to adapt scripts using the interface or similar interfaces, such as `ISenderReceiverInterface` or `IClientServerInterface`. Refer to the following listings that show the required script adaptations.

Old listing The following listing shows accessing the `IsService` property in prior SystemDesk versions.

```
# Setting the IsService property
interface.IsService = True

# Testing if the IsService property is specified
interface.IsIsServiceDefined

# Getting the IsService property
v = interface.IsService
```

New listing The following listing shows accessing the `IsService` property in SystemDesk 2023-B.

```
# Setting the IsService property
interface.SetNewIsService().SetValue(True)

# Testing if the IsService property is specified
interface.IsService is not None and
interface.IsService.IsCalculatedValueDefined

# Getting the IsService property
v = interface.IsService.CalculatedValue
```

Refer to `IPortInterface` in the *SystemDesk API Reference* that is only available in dSPACE Help.

Creating AUTOSAR elements

Creating AUTOSAR elements in a package changed for elements introduced with AUTOSAR R22-11. You can only generate the elements with generic methods that were added with this SystemDesk version. However, you do not have to migrate scripts creating AUTOSAR elements of AUTOSAR R21-11 or prior AUTOSAR versions.

Old script The `IPackageableElements` interface provides element-specific methods for creating AUTOSAR elements. Refer to the following listing for creating an `ICompuMethod` interface.

```
activeProject = systemDeskApplication.ActiveProject
root = activeProject.RootAutosar
packages = root.ArPackages
package = packages.Item("PackageableElements")
compuMethod = package.Elements.AddNewCompuMethod("myCompuMethod")
```

The automation interface provides the following element-specific properties and methods:

- `<Elements>`
- `AddNew<Element>(ShortName)`
- `AddNew<Element>Range(ShortName[])`

New script The following listing shows creating AUTOSAR elements with the generic methods that were added with this SystemDesk version. You can use the generic methods for all the AUTOSAR elements.

```
activeProject = systemDeskApplication.ActiveProject
root = activeProject.RootAutosar
packages = root.ArPackages
package = packages.Item("PackageableElements")
availableElements = package.Elements.AvailableElementTypes
for element in availableElements:
    myElement = package.Elements.AddNew(element, "my"+element)
```

The automation interface now provides the following generic methods:

- `AddNew(ElementType, ShortName)`
- `AddNewRange(ElementType, ShortName[])`
- `OfType(ElementType)`

The `IPackageableElements` interface additionally provides the `AvailableElementTypes` property that returns the interface names of all elements you can create in a package.

Note

You can use the following element names to specify the **ElementType** parameter of the **AddNew** and **AddNewRange** methods:

- Interface name, e.g., **ICompuMethod**
- Class name, e.g., **CompuMethod**
- XML name, e.g., **COMPU-METHOD**

Related information Refer to *IPackageableElements* in the *SystemDesk API Reference* that is only available in dSPACE Help.

Handling internal variables of SWC-internal behaviors

The handling of A2L variables you imported to SWC-internal behaviors changed with this version of SystemDesk. You can now import variables from A2L files to AUTOSAR static memories and constant memories of SWC-internal behaviors. In prior SystemDesk versions, the variables from A2L files were imported in a custom variable format.

You have to migrate scripts that automated the access to imported internal variables of SWC-internal behaviors and replace the old API with the API for the AUTOSAR model.

Old API The following API was removed from SystemDesk:

- *ISwcInternalBehavior::InternalVariables* property: The property for getting the internal variable collection <type *IInternalVariables*> of an SWC-internal behavior was removed.
- *IInternalVariables*: The collection with all properties and methods was removed from the SystemDesk API.
- *IInternalVariable*: The interface with all properties and methods was removed from the SystemDesk API.

API for the AUTOSAR model You can now use the following API for the AUTOSAR model:

- *ISwcInternalBehavior::ConstantMemories* property: Gets the list of Measurements <type *IParameterDataPrototypes*>.
- *IParameterDataPrototype*: Interface to a specific Measurement.
- *ISwcInternalBehavior::StaticMemories* property: Gets the list of Characteristics <type *IVariableDataPrototypes*>.
- *IVariableDataPrototype*: Interface to a specific Characteristic.

For details, refer to the *SystemDesk API Reference* that is only available in dSPACE Help.

TargetLink

New versioning scheme

Note

The versioning used for dSPACE products until now is changed to a calendar-based versioning scheme. Refer to *New versioning scheme* in [General Enhancements and Changes](#) on page 13.

Where to go from here

Information in this section

New Features of TargetLink 2023-B.....	180
Migrating to TargetLink 2023-B and TargetLink Data Dictionary 2023-B.....	199
Changes in Future TargetLink Versions.....	248

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 2023-B

Where to go from here

Information in this section

Modeling in Simulink or Stateflow.....	180
Adaptive AUTOSAR.....	185
Classic AUTOSAR.....	188
Code Generation Core Functionality.....	189
Model-in-the-Loop Simulation (MIL).....	190
Target Simulation (PIL).....	190
Usability.....	191
Code Generator Options.....	194
API Functions and Hook Scripts.....	195
Other.....	196

Modeling in Simulink or Stateflow

Where to go from here

Information in this section

Improved Array-of-Struct Support.....	181
Dynamic Element Specification.....	181
Solving Systems of Linear Equations	182
Support of Ensure Output is Virtual.....	183
Support of Variable Vector Widths for Prelookup Blocks.....	183
Working with Stateflow Objects.....	184
Improvements Concerning Function Interfaces.....	184

Improved Array-of-Struct Support

Support of array-of-bus signals

With Release 2023-B, TargetLink supports array-of-bus signals at the data inputs and outputs of the Switch block.

Related documentation

- [Basics on Array-of-Bus Signals and Array-of-Struct Variables \(TargetLink Preparation and Simulation Guide !\[\]\(8c4dca64662d21542001ca0ed7eeb688_img.jpg\)\)](#)

Access to arrays of structs using access functions

With TargetLink 2023-B, it is now possible to access array-of-struct variables using address access functions.

Related documentation

- [Determining Access Function Kind According to the Use Case \(TargetLink Customization and Optimization Guide !\[\]\(e548a391c65118ac2476924cdb5db38c_img.jpg\)\)](#)
- [Associating Access Functions With Variable Kinds \(TargetLink Customization and Optimization Guide !\[\]\(6fc1fda334fce799e3b50f6cf68d70a8_img.jpg\)\)](#)

Related topics

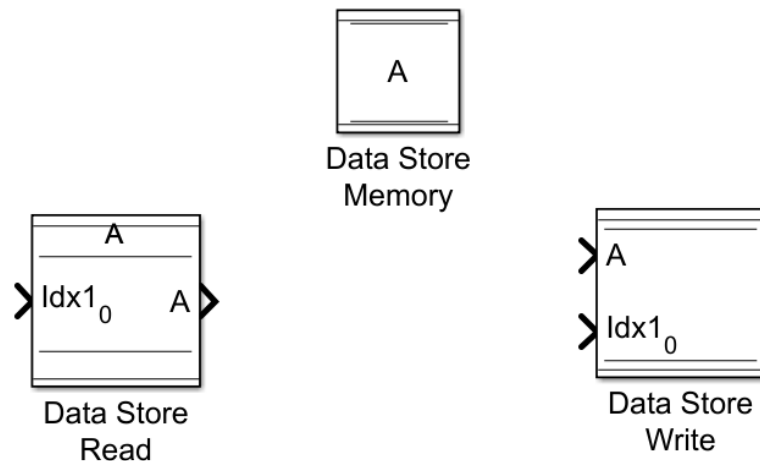
References

[Switch Block \(TargetLink Model Element Reference !\[\]\(bd3b31712ad9bab5a241210fa6925cdd_img.jpg\)\)](#)

Dynamic Element Specification

Dynamic element specification at Data Store Read and Data Store Write blocks

TargetLink 2023-B supports the Simulink Enable indexing option for element selection and assignment at Data Store Read and Data Store Write blocks. With this option, it is now possible to dynamically access variables, including array-of-struct variables.



This is equivalent to the same functionality realized by block ensembles with Data Store Read and Selector blocks or Data Store Read, Assignment, and Data Store Write blocks.

With TargetLink 2023-B, the TargetLink block dialogs of Data Store Read and Data Store Write blocks offer the possibility to switch easily to the appropriate Simulink block dialog for element specification using the Element Selection/Element Assignment buttons without unintentionally discarding data already entered.

Related documentation

- [How to Model the Access to Array-of-Struct Variables using Data Stores \(TargetLink Preparation and Simulation Guide !\[\]\(065aacad479feea1b3f501fa02b79a7a_img.jpg\)\)](#)
- [Basics on Dynamically Assigning New Values to Data Store Variables \(TargetLink Preparation and Simulation Guide !\[\]\(f90d8b6badff022f4fa9e71b17a20969_img.jpg\)\)](#)

Related topics

References

[Assignment Block \(TargetLink Model Element Reference !\[\]\(10f8862fc183b400327470ea85afe9ae_img.jpg\)\)](#)
[Data Store Read Block \(TargetLink Model Element Reference !\[\]\(4ba8d838a2aa5445d51c9dee78fcb0cc_img.jpg\)\)](#)
[Data Store Write Block \(TargetLink Model Element Reference !\[\]\(4fe307d00a844a23eb14d503e73187bd_img.jpg\)\)](#)

Solving Systems of Linear Equations

Solving systems of linear equations via decomposition

TargetLink 2023-B introduces the SOLE Solver block and the associated Linear Algebra Library for solving systems of linear equations (SOLE) via decomposition.

On principle, a system of linear equations (SOLE) can be solved via the matrix inverse. While mathematically elegant, this is not viable from a numerical point of view:

- Matrix inversion may not be possible or can be associated with rounding errors or overflow/underflow when using floating point arithmetic.
- To calculate the matrix inverse a SOLE, where the right side is the identity matrix, has to be solved (it might not have a solution for reasons presented in the first point).
- Using the matrix inverse to solve a SOLE (i.e., calculating the inverse matrix and then the solution) instead of using a decomposition usually results in a larger number of calculations.

Calculating the matrix inverse first and then multiplying it with the right side of the original SOLE is always one matrix multiplication more than solving the original SOLE directly.

- To solve a SOLE directly the left matrix of the SOLE is decomposed into matrices with special properties, which makes the SOLE easy to solve.

To facilitate precision, TargetLink solves SOLE via decomposition.

As achieving a valid decomposition involves similar problems, TargetLink provides an error status that you can use to respond to numerical problems.

Related documentation

- [Solving Systems of Linear Equations \(TargetLink Preparation and Simulation Guide !\[\]\(36f8637baaa56c4be44b454435949289_img.jpg\)\)](#)
- [SOLE \(TargetLink Demo Models !\[\]\(b556e0ef1e10ccfc32976edb6416074f_img.jpg\)\)](#)
- [SOLE Solver Block \(TargetLink Model Element Reference !\[\]\(cf1529ba638f0498d7e334e7a79dd058_img.jpg\)\)](#)

Related topics**References**

[SOLE Solver Block Functionality \(TargetLink Model Element Reference !\[\]\(83f22ed94ec5517769dd76d702c6bfd8_img.jpg\)\)](#)

Support of Ensure Output is Virtual

Support of the Ensure output is virtual parameter

With Release 2023-B, TargetLink supports the Simulink® parameter Ensure output is virtual at the unenhanced Output blocks of conditionally executed subsystems.

Switching from classic initialization mode to simplified initialization mode by using the Simulink diagnostics parameter Underspecified initialization detection can cause differences in MIL simulation and in the generated code. For example, additional IF variables are added.

To avoid these differences in simplified initialization mode, you can select the Ensure output is virtual checkbox in the Simulink dialog of the respective Output block.

Related documentation

- [Ensure Output is Virtual \(Limitations\) \(TargetLink Limitation Reference !\[\]\(1f101ad452ef9a3f01bb1e89af34fc34_img.jpg\)\)](#)
- [Simulink documentation by MathWorks®](#)

Support of Variable Vector Widths for Prelookup Blocks

Using width macros

With TargetLink 2023-B, the Prelookup block supports a variable vector width implemented as width macro for the Axis/Breakpoint variable.

Related documentation

- [Making Variables Width-Varying \(TargetLink Customization and Optimization Guide !\[\]\(8134b5476de36b4fd1f82e8a3cb19300_img.jpg\)\)](#)
- [Example of Using Width-Varying Variables in the Context of Prelookup Blocks \(TargetLink Customization and Optimization Guide !\[\]\(8bcb365fec0707588379d7b91fa971fa_img.jpg\)\)](#)

Related topics**References**

[Prelookup Block \(TargetLink Model Element Reference !\[\]\(3d8c13c92b853674f749aac6fa869926_img.jpg\)\)](#)

Working with Stateflow Objects

Open Stateflow dialogs

With TargetLink 2023-B, you can open Stateflow® object dialogs directly from the model by using the Open Stateflow Dialogs dialog in the TargetLink tab on the Simulink® Toolstrip.

Related documentation

- [Basics on Working with Stateflow Objects \(TargetLink Preparation and Simulation Guide !\[\]\(0f48f43ebd21f231a458c96216dbf4d1_img.jpg\)\)](#)

Improvements Concerning Function Interfaces

MISRA-compliant modeling at function interfaces

With TargetLink 2023-B, you can qualify value or reference function parameters at function interfaces as `const`. The predefined system templates provide the following preconfigured DD VariableClass objects for this purpose:

- FCN_INPUT
- FCN_REF_INPUT

This lets you model in line with MISRA.

The following model elements support this modeling approach:

- InPort block (in non-virtual subsystems)
- Bus Inport block (in non-virtual subsystems)
- ArgIn block
- Data objects of a Stateflow chart with scope Input except inputs of Stateflow MATLAB Functions.

Related documentation

- [Basics on Function Interfaces \(TargetLink Customization and Optimization Guide !\[\]\(3403f05cd757a0fd15a71dc598e177cd_img.jpg\)\)](#)
- [Details on Preconfigured VariableClass Objects in the Data Dictionary \(TargetLink Customization and Optimization Guide !\[\]\(3d09655b25e2951ebcfe2fcf5572c2f9_img.jpg\)\)](#)

Related topics**References**

[ArgIn Block \(TargetLink Model Element Reference !\[\]\(666e09182d4cd268646ea700ea60dcdf_img.jpg\)\)](#)
[Bus Inport Block \(TargetLink Model Element Reference !\[\]\(1ef1ef0bf9af6c6996401964cf280f2d_img.jpg\)\)](#)
[InPort Block \(TargetLink Model Element Reference !\[\]\(e9a80c8557f9285916925bd4ac40fff5_img.jpg\)\)](#)

Adaptive AUTOSAR

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

Supported Adaptive AUTOSAR Releases.....	185
Adaptive AUTOSAR Frame Model Generation	185
C++ Enumeration Class Data Types.....	186
New Adaptive AUTOSAR Modeling Features.....	186
Changes According to AUTOSAR Version R22-11.....	187
C++ Namespaces.....	187

Supported Adaptive AUTOSAR Releases

TargetLink 2023-B now supports the following Adaptive AUTOSAR releases:

- R22-11
- R21-11
- R20-11

Adaptive AUTOSAR Frame Model Generation

Generating Adaptive AUTOSAR frame models

With TargetLink 2023-B, you can now generate Adaptive AUTOSAR frame models from AUTOSAR specifications in the Data Dictionary. The starting point is an ARXML file containing your Adaptive AUTOSAR specification. After importing the specification to the Data Dictionary, you can generate an Adaptive AUTOSAR frame model. TargetLink offers you different possibilities to customize the Adaptive AUTOSAR frame model generation.

Related documentation

- [Basics on Adaptive AUTOSAR Frame Models \(TargetLink Adaptive AUTOSAR Modeling Guide !\[\]\(95b42f0077faf7439a26242a54e021ec_img.jpg\)\)](#)
- [How to Start the Generation of a Frame Model from Adaptive AUTOSAR Data \(TargetLink Adaptive AUTOSAR Modeling Guide !\[\]\(e097ab4c08b8186dd0908330bbc2dc28_img.jpg\)\)](#)

C++ Enumeration Class Data Types

C++ Enumeration class data types

With TargetLink 2023-B, you can now use C++ enumeration class data types at Adaptive AUTOSAR interfaces.

Related documentation

- [Basics on Working with Enumeration Data Types in TargetLink \(TargetLink Preparation and Simulation Guide !\[\]\(c6a8736a601a632e2c96605cf66055ed_img.jpg\)\)](#)
- [Basics on Data Types in Adaptive AUTOSAR \(TargetLink Adaptive AUTOSAR Modeling Guide !\[\]\(64ef2b19d70b31fbbfce0e0e2aa3d7b4_img.jpg\)\)](#)

New Adaptive AUTOSAR Modeling Features

Support of user-specified data types when modeling access to persistent memory

In addition to AdaptivePlatformTypes, TargetLink 2023-B also supports user-specified data types (e.g., scalars, structures, arrays, and enumerations) when modeling the access to persistent memory.

Related documentation

- [Accessing Persistent Memory \(TargetLink Adaptive AUTOSAR Modeling Guide !\[\]\(609f3372828e3526d7ce4ba9a1b5248e_img.jpg\)\)](#)

Support of data store blocks for events and fields

As of TargetLink 2023-B, you can now access fields or receive/send events using data store blocks.

Related documentation

- [How to Receive Events \(Adaptive AUTOSAR\) \(TargetLink Adaptive AUTOSAR Modeling Guide !\[\]\(3dc92c626ede9fa1b47e2e010104b5c4_img.jpg\)\)](#)
- [How to Send Events \(Adaptive AUTOSAR\) \(TargetLink Adaptive AUTOSAR Modeling Guide !\[\]\(71e9a2c5583c3d2a2fe005f4239e5d39_img.jpg\)\)](#)
- [How to Get or Set the Value of a Field as a Service Consumer \(TargetLink Adaptive AUTOSAR Modeling Guide !\[\]\(5aa86d4510bccda8326cff2d2412c12b_img.jpg\)\)](#)
- [How to Update the Value of a Field as a Service Provider \(TargetLink Adaptive AUTOSAR Modeling Guide !\[\]\(1ed15c9cc7c61868b75e5155e86ba5c0_img.jpg\)\)](#)

Changes According to AUTOSAR Version R22-11

New API: `GetServiceState()`

With TargetLink 2023-B and Adaptive AUTOSAR version R22-11, the new API function `GetServiceState()` is used in the ARA adapter code. It provides information on the service state, i.e., based on the return value, you can determine if a service is available. This API function is not used in the ARA adapter code for Adaptive AUTOSAR Versions < R22-11.

C++ Namespaces

Addressing namespaces to symbols in modules and software components

As of TargetLink 2023-B, you can address a namespace to symbols within a module or software component. This means that the namespace is then inherited by all functions, variables, and data types within the software component or module which do not address a namespace on their own.

Related documentation

- [Basics on Using Namespaces \(TargetLink Customization and Optimization Guide !\[\]\(e492b5d52ab457a7a3c2826c4091dfee_img.jpg\)\)](#)

Influencing namespace qualification

With TargetLink 2023-B, you can configure the Code Generator to either qualify all symbols with their full or reduced namespace. If you set the `NamespaceQualification` Code Generator option to `Reduced`, the symbols are only qualified with the namespace that differs from the surrounding namespace.

Related documentation

- [NamespaceQualification \(TargetLink Model Element Reference !\[\]\(69baca079ef3ab6f03d58fd7e9f950f1_img.jpg\)\)](#)
- [Basics on Using Namespaces \(TargetLink Customization and Optimization Guide !\[\]\(2da321c3dc978a55192cb9c452297973_img.jpg\)\)](#)

Related topics

References

[NamespaceQualification \(TargetLink Model Element Reference !\[\]\(2bae76de5ebbd5c4d7d47162f1673734_img.jpg\)\)](#)

Classic AUTOSAR

Where to go from here

Information in this section

Supported Classic AUTOSAR Releases.....	188
Improved Array-of-Struct Support (Classic AUTOSAR).....	188
AUTOSAR Import and Export Improvements.....	188

Supported Classic AUTOSAR Releases

TargetLink 2023-B now supports Classic AUTOSAR Release R22-11.

For a list of all the supported Classic AUTOSAR Releases, refer to [Supported AUTOSAR Releases \(TargetLink Interoperation and Exchange Guide !\[\]\(ec9132f1d27c8919987d92907322654d_img.jpg\)](#)).

Improved Array-of-Struct Support (Classic AUTOSAR)

Improved ARXML import/export and code generation

With TargetLink 2023-B, the ARXML import/ export and code generation were improved for data prototypes typed by array of structs. The ARXML import/export, and code generation are now possible for the following data prototypes:

- PerInstanceMemory
- SharedCalPrm
- PerInstanceCalPrm
- CalPrmElement

AUTOSAR Import and Export Improvements

Support of parameter SWCs during import

With TargetLink 2023-B, you can now import ARXML files containing parameter SWCs to the Data Dictionary. During AUTOSAR frame model generation, these parameter SWCs are not taken into account.

Related documentation

- [Basics on Importing AUTOSAR Files \(TargetLink Interoperation and Exchange Guide !\[\]\(3292f5442e3b4027aa0bb60988f9fc82_img.jpg\)](#))

New property values for DD SwRecordLayoutContent objects

TargetLink 2023-B offers the following new property values for the Kind property of DD SwRecordLayoutContent objects:

- RESERVED
- IDENTIFIER

As of AUTOSAR Release R22-11, both property values are taken into account during AUTOSAR import/export.

Related documentation

- [SwRecordLayoutContentKindEnum](#)

Code Generation Core Functionality

Improved Code Efficiency

Struct component sorting

With TargetLink 2023-B, you can instruct TargetLink to sort the components of struct types that are specified via structured DD Typedef objects. The sorting order is the same as the default for the sorting of components of implicitly generated struct types and is intended to achieve a small memory footprint for variables of this type or incorporating this type as component.


Related documentation

- [Optimization \(Typedef Object\)](#)
- [Changed sorting order of struct components of implicit structs](#) on page 233
- [Sorting of struct components for implicitly generated bus structs](#) on page 234

Duplication of function calls for stack/code size tradeoff

By effectively moving code to mutually exclusive control flow branches, expensive intermediate variable copies can be avoided at the cost of multiple instances of the code.

Related documentation

- [UtilizeCodeDuplicationInControlFlowBranches](#) (TargetLink Model Element Reference )
- [Code duplication in conditionally executed control flows](#) on page 225

Model-in-the-Loop Simulation (MIL)

Analyzing Simulation Results

Improved plot zooming

With TargetLink 2023-B, the plot zooming in the More Infos Plot Window is improved. The x-axis scalings of the signal plot and the Deviation to other signals in Detailed Plot are always synchronous.

Related documentation

- [Example of Investigating Quantization Errors \(TargetLink Preparation and Simulation Guide !\[\]\(444b1eae2189e5cd8d096594c07a0a6e_img.jpg\)](#))

Target Simulation (PIL)

Changes in the Target Simulation Modules with TargetLink 2023-B

New and discontinued compiler versions

The following table shows changes in the compiler versions that are supported by TargetLink 2023-B:

Microcontroller Family	Compiler	New Compiler Version	Discontinued Compiler Version
MPC57xxVLE	GreenHill	2023	2022
RH850	GreenHill	2023	2022
TriCore4xx	SMARTCODE	10.1	—

For a complete list of evaluation boards and compilers that are supported by TargetLink, refer to [Combinations of Evaluation Boards and Compilers \(Evaluation Board Reference !\[\]\(ab4e2b3fc7e7887b7a72f548aa6f5e60_img.jpg\)](#)).

Note

For more PIL support combinations that are part of a valid Software Maintenance Service (SMS) contract, refer to the dSPACE [TargetLink PIL Support](#) website at the [TargetLink Product Support Center](#).

Usability

Usability Improvements

Improvements of the DD three-way merge

TargetLink 2023-B provides several usability improvements concerning the DD three-way merge, such as a new filter to show only conflicting DD objects.

Related documentation

- [How to Perform a DD Three-Way Merge \(TargetLink Data Dictionary Basic Concepts Guide !\[\]\(5774573cf757c446bb08af21f46b2969_img.jpg\)\)](#)
- [DD Three-Way Merge \(TargetLink Data Dictionary Manager Reference !\[\]\(a502cb21d600ba28a5cdf414d68eef89_img.jpg\)\)](#)

Support of a text-based DD file

TargetLink 2023-B lets you save DD files as `.dd` or `.ddjson` files. The text-based `.ddjson` format lets you use DD files in version control systems like Git, to do the following:

- Display changes in a DD project file.
- Perform reviews on DD project file contents.
- Use the blame function.

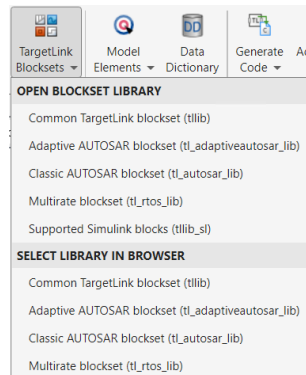
Related documentation

- [Basics on DD File Formats \(TargetLink Data Dictionary Basic Concepts Guide !\[\]\(029651ce9ee64da8525b17c64e266edc_img.jpg\)\)](#)
- [Data Dictionary Command Line Tool API Command \(TargetLink Data Dictionary Manager Reference !\[\]\(05d3bfcaecedf25939aadd260bd34af7_img.jpg\)\)](#)

Improvements of the TargetLink tab on the Simulink Toolstrip

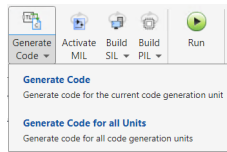
TargetLink 2023-B provides the following usability improvements of the TargetLink tab on the Simulink[®] Toolstrip:

Library section



The extended Library section offers the new TargetLink Blocksets command including shortcuts to the TargetLink libraries with the choice of opening them or displaying them in the Simulink[®] Library Browser.

Generate Code

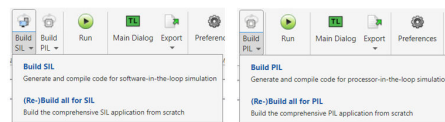


The Generate Code command in the PRODUCTION CODE section now offers the following options:

- Clicking the icon or Generate Code in the pop-up list starts the code generation only for the current code generation unit (CGU).
- Clicking Generate Code for all Units in the pop-up list starts the code generation for the current subsystem or the corresponding code generation unit (CGU) including subitems.

Refer to [Overview on Starting Code Generation \(TargetLink Preparation and Simulation Guide\)](#).

Build SIL/Build PIL



In the PRODUCTION CODE section, also the Build SIL and Build PIL commands were extended:

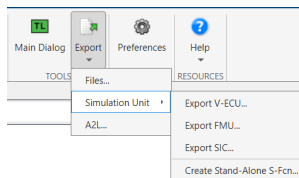
- Clicking the icon or Build SIL/Build PIL in the pop-up list starts the respective build process for the current TargetLink subsystem. As first step, the code for the current subsystem or the corresponding code generation unit (CGU) including subitems is generated.

Refer to [How to Start the Build Process for Selected Subsystems \(TargetLink Preparation and Simulation Guide\)](#).

- Clicking (Re-)Build all for SIL/(Re-)Build all for PIL in the pop-up list starts the build process for all your TargetLink subsystems in the model in one step.







Refer to [How to Start the Build Process for All Subsystems \(TargetLink Preparation and Simulation Guide\)](#).

Tools section



The new Tools section includes the following two commands:

- Clicking Main Dialog opens the TargetLink Main Dialog.
- Clicking Export and selecting an item of the pop-up list opens the corresponding TargetLink utility or, in case of A2L, starts the file generation. The following commands are available:

Command	Description	Documentation
Files...	Opens the TargetLink File Export Utility.	TargetLink File Export Utility (TargetLink Tool and Utility Reference )
Simulation Unit - Export V-ECU...	Opens the TargetLink V-ECU Manager.	TargetLink V-ECU Manager (TargetLink Tool and Utility Reference )
Simulation Unit - Export FMU...	Opens the TargetLink FMU Manager.	TargetLink FMU Manager (TargetLink Tool and Utility Reference )
Simulation Unit - Export SIC...	Opens the TargetLink SIC Manager.	TargetLink SIC Manager (TargetLink Tool and Utility Reference )
Simulation Unit - Create Stand-Alone S-Fcn...	Opens the TargetLink Stand-Alone Model Manager.	Stand-Alone Model Manager (TargetLink Tool and Utility Reference )
A2L	Starts the ASAM MCD-2 MC file generation.	dsdd_export_a2l_file (TargetLink API Reference )

Improved Autoscaling View Set

With TargetLink 2023-B, the Autoscaling View Set of the Property Manager additionally shows the Min and Max columns.

Related documentation

- [How to Optimize the Property View for Your Use Cases \(TargetLink Preparation and Simulation Guide !\[\]\(30a147af384f9f71632c2ff17bc706c8_img.jpg\)\)](#)
- [View Set Chooser \(TargetLink Tool and Utility Reference !\[\]\(9b33568d5c136f08ca688ce48be37574_img.jpg\)\)](#)

Related topics

References

[TargetLink Main Dialog Block \(TargetLink Model Element Reference !\[\]\(17413706fd4997a1a4bdf85c6864eee1_img.jpg\)\)](#)

Code Generator Options

New Code Generator Options

The following new Code Generator options are available with TargetLink 2023-B:

- **DecomposeStructAssignmentsForInitialValuePropagation**
Lets you replace struct assignments with component-wise assignments, if this allows the replacement of the right-side component accesses with their respective initial values.
- **NamespaceQualification**
Specifies whether symbols in the generated C++ code are qualified with their fully qualified namespace or only with the part that differs from the surrounding namespace.
- **UtilizeCodeDuplicationInControlFlowBranches**
By effectively moving code into mutually exclusive control flow branches, expensive intermediate variable copies can be avoided at the cost of multiple instances of the code.
- **EmitWarningIfMathHeaderIsIncluded**
Specifies whether a warning is emitted if the C standard library header `math.h` is included.
- **UseTLImplementationForFabsFunction**
Specifies whether `fabs` function from the C standard library header `math.h` is implemented as a conditional expression using the `?` operator directly in the production code.

Related documentation

- [Basics on Using Namespaces \(TargetLink Customization and Optimization Guide !\[\]\(4e333a6106fc298d0ae6dff272a736ef_img.jpg\)\)](#)

Migration aspects of Code Generator options

For more information, refer to [Migration Aspects Regarding Code Generator Options](#) on page 208.

Related topics

References

[DecomposeStructAssignmentsForInitialValuePropagation \(TargetLink Model Element Reference !\[\]\(6a9b39b98eb945faa14c645ec99e4eaa_img.jpg\)\)](#)
[EmitWarningIfMathHeaderIsIncluded \(TargetLink Model Element Reference !\[\]\(182077db5bac9ff62bf376fe37ffa951_img.jpg\)\)](#)
[NamespaceQualification \(TargetLink Model Element Reference !\[\]\(6ed6a340e0627314752774197e63f07e_img.jpg\)\)](#)
[UseTLImplementationForFabsFunction \(TargetLink Model Element Reference !\[\]\(9ff6d055737e0f9bee95f39dc7b63bda_img.jpg\)\)](#)
[UtilizeCodeDuplicationInControlFlowBranches \(TargetLink Model Element Reference !\[\]\(d6c755287beca994c9f500381644d155_img.jpg\)\)](#)

API Functions and Hook Scripts

Where to go from here

Information in this section

New API Functions.....	195
New Hook Scripts.....	195

New API Functions

API Function	Purpose
tlExportCLibrary	Exports a TargetLink C library as a package container or as a ZIP file.
tlRebuildLinearAlgebraLibrary	Rebuilds the TargetLink Linear Algebra Library.

Related topics

References

[tlExportCLibrary \(TargetLink API Reference !\[\]\(5361750c22c4e047a52f4eac1ec2d4cc_img.jpg\)\)](#)
[tlRebuildLinearAlgebraLibrary \(TargetLink API Reference !\[\]\(f276343e5e0d2402c20fdc9e8443c0dd_img.jpg\)\)](#)

New Hook Scripts

Hook Script	Description
tl_post_add_adaptiveearfunctions subsystem_hook	Lets you enter your own commands to be executed after a Adaptive AUTOSAR Function subsystem is added to the frame model.
tl_post_add_methodbehaviorsubsystem_hook	Lets you enter your own commands to be executed after a method behavior subsystem is added to the frame model.

Hook Script	Description
tl_post_add_methodcallsubsystem_hook	Lets you can enter your own commands to be executed after a method call subsystem is added to the frame model.

Related topics

References

[tl_post_add_adaptivearfunctions subsystem_hook \(TargetLink File Reference !\[\]\(e78f798d4ea5c530c9db49e7d26e6b95_img.jpg\)\)](#)
[tl_post_add_methodbehaviorsubsystem_hook \(TargetLink File Reference !\[\]\(034433b90593e82e5460e34e3ed48e9b_img.jpg\)\)](#)
[tl_post_add_methodcallsubsystem_hook \(TargetLink File Reference !\[\]\(5f24500834b50a8307ffe63e419281a9_img.jpg\)\)](#)

Other

Where to go from here

Information in this section

[FMI 3.0 Support..... 196](#)
[TargetLink Demos and Examples..... 197](#)

FMI 3.0 Support

TargetLink supports FMI 3.0

TargetLink 2023-B supports the following features of the FMI 3.0 standard:

- Co-Simulation interface
- Integer, boolean, float (32- and 64-bit), and enumeration data types
- Multi-dimensional variables (e.g., vectors and matrices)

Related documentation

- [Basics on Exporting FMUs from TargetLink \(TargetLink Interoperation and Exchange Guide !\[\]\(f6ec0429e2f15dbd4a39b612501b6fa4_img.jpg\)\)](#)

TargetLink Demos and Examples

New demo models


New Demo Model	Description
TABLEND	The <code>tableNd</code> demo model shows you available methods to specify the table values of look-up tables up to five dimensions. Refer to Introduction to Look-up Tables (TargetLink Preparation and Simulation Guide) .
SOLE	This demo shows how to solve systems of linear equations with the SOLE Solver block using the Linear Algebra Library. Refer to Solving Systems of Linear Equations (TargetLink Preparation and Simulation Guide) .

Changed demo models

Changed Demo Model	Description
AAR_COMMUNICATION	A C++ enumeration class data type is used at the RP_ServiceB FieldB InPort block. In addition, the software component is assigned via the GeneralNamespaceRef property to the namespace <code>dspace/targetlink</code> . This means that the namespace is inherited by all functions, variables, and types in the software component which do not specify another namespace on their own.

Removed demo models

Removed Demo Model	Description
MATRIX_INVERSE	<p>On principle, a system of linear equations (SOLE) can be solved via the matrix inverse. While mathematically elegant, this is not viable from a numerical point of view:</p> <ul style="list-style-type: none"> Matrix inversion may not be possible or can be associated with rounding errors or overflow/underflow when using floating point arithmetic. To calculate the matrix inverse a SOLE, where the right side is the identity matrix, has to be solved (it might not have a solution for reasons presented in the first point). Using the matrix inverse to solve a SOLE (i.e., calculating the inverse matrix and then the solution) instead of using a decomposition usually results in a larger number of calculations. <p>Calculating the matrix inverse first and then multiplying it with the right side of the original SOLE is always one matrix multiplication more than solving the original SOLE directly.</p>

Removed Demo Model	Description
	<ul style="list-style-type: none">▪ To solve a SOLE directly the left matrix of the SOLE is decomposed into matrices with special properties, which makes the SOLE easy to solve. <p>To facilitate precision, TargetLink solves SOLE via decomposition.</p> <p>Accordingly, the <code>MATRIX_INVERSE</code> demo was removed from TargetLink. It is replaced by the SOLE demo.</p> <p>Refer to Solving Systems of Linear Equations (TargetLink Preparation and Simulation Guide ).</p>

Related topics

Basics

[AAR_COMMUNICATION \(TargetLink Demo Models !\[\]\(d3fb9f94af8b26d1c844efa9a98805b0_img.jpg\)](#))

[SOLE \(TargetLink Demo Models !\[\]\(950a62bbddad88d64435fd35607dfc42_img.jpg\)](#))

[TABLEND \(TargetLink Demo Models !\[\]\(5a132f13505a6571904d622757b7a8f0_img.jpg\)](#))

References

[SOLE Solver Block Functionality \(TargetLink Model Element Reference !\[\]\(e1d6102fe77919492c04879c8450f1f5_img.jpg\)](#))

Migrating to TargetLink 2023-B and TargetLink Data Dictionary 2023-B

Where to go from here

Information in this section

General Migration Information.....	199
Migrating from TargetLink 2022-B to 2023-B.....	207
Code Changes Between TargetLink 2022-B and TargetLink 2023-B.....	217
Discontinuations as of TargetLink 2023-B.....	245

General Migration Information

Where to go from here

Information in this section

Upgrading Models, Libraries, and Data Dictionaries.....	199
Changes in Rounding Behavior for Exactly Representable Floating-Point Numbers (Microsoft).....	207

Upgrading Models, Libraries, and Data Dictionaries

Where to go from here

Information in this section

Basics on Migrating Between TargetLink Versions.....	200
How to Upgrade and Save a Data Dictionary with Included DD Files.....	202
How to Manually Upgrade Libraries and Models Using the API.....	203
Migrating Data Dictionaries to CodeDecorationSets.....	204

Basics on Migrating Between TargetLink Versions

Automatic upgrade from TargetLink 3.1 or later

TargetLink 2023-B automatically upgrades models and TargetLink-compliant libraries if they were created with TargetLink 3.1 or later.

The automatic upgrade includes all steps required by the intermediate TargetLink versions. Internally, the model and the Data Dictionary are upgraded automatically step-by-step from version to version.

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.
- 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). Refer to [Basics on Code Formatting \(TargetLink Customization and Optimization Guide\)](#).
- Custom code S-functions built with 32-bit TargetLink versions do not work with 64-bit versions of TargetLink.

Initiate a rebuild of all custom code S-functions using the `tlUpgrade('Model', <MyModel>, 'CheckModel', 'FixIssues')` API function.

- Older models use the predefined DD EnumImplementation object for Simulink enumeration data types to generate integer values of the Simulink enumeration constants in the production code. Replace this data type with the predefined IMPLICIT_ENUM data type to generate implicit TargetLink enumeration data types in accordance with the Simulink enumeration data type. Refer to [Basics on Working with Enumeration Data Types in TargetLink \(TargetLink Preparation and Simulation Guide\)](#).

Upgrading Data Dictionaries

If you open a DD file that was last saved with an earlier TargetLink version, it must be upgraded. The upgrade is executed automatically. The Subsystems area and the <Application> area are no longer compatible and thus removed. To change the upgrade behavior, refer to `tl_pref` ([TargetLink API Reference](#)).

If you save the upgraded DD file, the original DD file is automatically saved to a backup DD file in the same directory as the DD file before it is overwritten with the upgraded version. To disable the automatic backup, refer to [Data Dictionary Upgrade in Topic Navigator \(TargetLink Tool and Utility Reference\)](#).

To upgrade DD files with included DD files, refer to [How to Upgrade and Save a Data Dictionary with Included DD Files](#) on page 202.

Changes in DD predefined system templates

With a new TargetLink version, the content of the DD predefined system templates might have been changed. To visualize these changes, you can compare your DD project file with the corresponding DD template of the current TargetLink version. If required, you can merge new or changed DD objects in your DD project file. Refer to [New - Create New DD Workspace \(TargetLink Data Dictionary Manager Reference !\[\]\(919a2cb85b99741a73c0c31a427236a8_img.jpg\)\)](#) and [Basics on Comparing DD Objects and Workspaces Using the DD Two-Way Comparison \(TargetLink Data Dictionary Basic Concepts Guide !\[\]\(c9cd5a1c35167a83f09a35036fe5dcbd_img.jpg\)\)](#).

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 automatically become a TargetLink library if it contains TargetLink blocks. The library itself must be TargetLink-compliant.

Refer to [How to Make User Libraries Upgrade-Capable \(TargetLink Orientation and Overview Guide !\[\]\(003082e50e3009141f59bd5df831749f_img.jpg\)\)](#).

Making existing libraries TargetLink-compliant

The following two approaches let you make libraries created with earlier TargetLink versions compliant with TargetLink 2023-B:

The earlier TargetLink version is available Use the TargetLink version with which the library was created to make the library TargetLink-compliant. Refer to the TargetLink documentation of the earlier TargetLink version. You can then use this library with all later TargetLink versions because TargetLink automatically performs an upgrade. The library can still be used with TargetLink versions earlier than TargetLink 2023-B because the automatic upgrade does not save a library in the newer TargetLink version.

Only the current TargetLink version 2023-B is available Use TargetLink 2023-B and the `t1Upgrade` API command to make the library TargetLink-compliant. Refer to [How to Manually Upgrade Libraries and Models Using the API](#) on page 203. If you follow the instructions, the library is saved in TargetLink 2023-B. Therefore, it cannot be used with TargetLink versions earlier than TargetLink 2023-B.

Manual upgrade from TargetLink 2.x or 3.0.x

Models and libraries created with TargetLink versions 2.x or 3.0.x have to be upgraded manually to the latest TargetLink version 3.x (3.1...3.5) you have. Afterwards, automatic upgrade is possible.

No downward compatibility

You cannot use models, libraries, or Data Dictionaries in the format of newer TargetLink versions in earlier TargetLink versions.

Data model filter rule files

Existing data model filter rule files can contain invalid elements because the data model of the TargetLink Data Dictionary changed. The following files that were shipped with previous TargetLink versions can be affected:

- DD_Filter_Admin.xml
- DD_Filter_AR_User.xml
- DD_Filter_NonAR_NonRTOS_User.xml

You can check filter rule files using 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 any XML-capable editor:

- File name
- Row number
- Column number

Related topics

Basics

- [Basics on Code Formatting \(TargetLink Customization and Optimization Guide !\[\]\(b6d55d0b173caf9b2505126db01e6158_img.jpg\)](#))
- [Basics on Filter Rule Sets for the Data Model \(TargetLink Data Dictionary Basic Concepts Guide !\[\]\(12811766810e4126d2bed4d8c0808e60_img.jpg\)](#))

HowTos

- [How to Make User Libraries Upgrade-Capable \(TargetLink Orientation and Overview Guide !\[\]\(adb0331d22f78481623cc605df40612a_img.jpg\)](#))
- [How to Manually Upgrade Libraries and Models Using the API..... 203](#)

References

- [tlUpgrade \(TargetLink API Reference !\[\]\(5a0d662075632df1b39c9e3427a70093_img.jpg\)](#))

How to Upgrade and Save a Data Dictionary with Included DD Files

Precondition

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

Method**To upgrade and save a Data Dictionary with included DD files**

- 1 Open the Data Dictionary Manager and the main DD file using File - Open. The main DD file and all included DD files are automatically upgraded to the latest data model revision.
- 2 Set the AutoSave property of all desired `/Config/DDIncludeFiles/<DDIncludeFile>` objects to On to save the included DD files together with the main DD file.
- 3 Save the main DD file using File - Save.

Result

You upgraded and saved the main DD file and all included DD files. TargetLink adjusted the revision number of every upgraded DD file to the latest data model revision. Backups of the original DD files are also saved to the same directory as the DD file.

Related topics**Basics**

[Basics on Opening and Handling DD Files \(TargetLink Data Dictionary Basic Concepts Guide !\[\]\(3e2231b1ad3ca8da8658228c00dd08e0_img.jpg\)\)](#)

HowTos

[How to Include Partial Data Dictionary Files \(TargetLink Data Dictionary Basic Concepts Guide !\[\]\(870f5d5e9c0d57485634be3ecf52f3ca_img.jpg\)\)](#)

References

[Point of Inclusion \(TargetLink Data Dictionary Manager Reference !\[\]\(0d5ec72f61334709c3fc9450209b754f_img.jpg\)\)](#)

How to Manually Upgrade Libraries and Models Using the API

Objective

To prepare a central upgrade of libraries and models in a tool chain 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 DD project file has been opened and automatically upgraded, for example, using `dsdd_manage_project('Open', '<name>.dd')` or `dsdd('Open', '<name>.dd')`.

Method**To manually upgrade libraries and models using the API**

1 Type the following API command in the MATLAB Command Window:

```
tlUpgrade('Model', '<Model|Library>.slx',
'CheckModel', 'FixIssues')
```

The model or library is upgraded.

Note

When you upgrade models and libraries, first upgrade models or libraries that do not reference any other libraries, i.e., the libraries whose blocks and subsystems 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.slx`.

3 Repeat steps 1 and 2 for all other models and libraries.

Result

You upgraded the 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 VariableClass	Removal of the DeclarationStatements and SectionName properties.	The DeclarationStatements and SectionName properties of the DD CodeDecoration.Settings object.
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 you open a Data Dictionary whose data model is older than the latest revision, TargetLink prompts you to perform an automatic upgrade.

Object Kind	Trigger	Upgrade Action
VariableClass FunctionClass	DeclarationStatements or SectionName properties are set.	<ol style="list-style-type: none"> 1. Create a DD CodeDecorationSet object. 2. Create 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. Reference the CodeDecorationSet object at the original object.
SubStructTemplate	Filter.VariableClass is set.	<p>Transfer the values of the following properties from the variable class to the SubStructTemplate object's filter:</p> <ul style="list-style-type: none"> ▪ DeclarationStatements ▪ SectionName ▪ TypePrefix
VariableClassTemplate	<ul style="list-style-type: none"> ▪ Filter.FilterCondition is set to ALL_TRUE. ▪ Settings.VariableClass references a DD VariableClass object whose DeclarationStatements or SectionName property is 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. 	<ol style="list-style-type: none"> 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 DD VariableClassTemplate object with the same value at the Filter.VariableClassSpec property, add a DD CodeDecoration object to the CodeDecorationSet object. 4. Specify the DD CodeDecoration object as required. 5. Reference the DD CodeDecorationSet object at the VariableClass object created in step 1. 6. Reference the DD VariableClass object created in step 1 by 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 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 previous Data Dictionary. You can clean it manually to reduce the number of objects in the new Data Dictionary.

Merging width-specific variable classes If the previous Data Dictionary contained width-specific VariableClassTemplate/VariableClass objects, the 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 `<Name>` variable class 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 using its filter.
 3. Reference the resulting `CodeDecorationSet` object at the `<Name>` variable class.
- 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`.
 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 previous width-specific variable classes, you can use them again as the target of the 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 DD CodeDecoration objects can influence the generated production code 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 \(TargetLink New Features and Migration Guide 📖\)](#).

Changes in Rounding Behavior for Exactly Representable Floating-Point Numbers (Microsoft)

Changes in Rounding Behavior for Exactly Representable Floating-Point Numbers (Microsoft)

Introduction

Microsoft changed the rounding behavior of the `printf` family of functions. Starting in Windows 10 version 2004 (build 19041), these functions now print exactly representable floating-point numbers as described in [IEEE 754-2019](#).

Impact on code generation with TargetLink

This change made by Microsoft impacts code generation with TargetLink. Depending on the Windows version, the rounding behavior can differ between machines. Note that this is independent of TargetLink.

Windows 10 < 2004 (build 19041)	Windows 10 ≥ 2004 (build 19041)
Exactly representable floating-point numbers ending in 5 are rounded up.	Banker's rounding as described in IEEE 754-2019 .

Migrating from TargetLink 2022-B to 2023-B

Where to go from here

Information in this section

Code Generator Options.....	208
API Functions and Hook Scripts.....	210

Other Migration Aspects.....	215
MATLAB Support.....	216

Code Generator Options

Migration Aspects Regarding Code Generator Options

ConformingFreestandingImplementation

The ConformingFreestandingImplementation Code Generator option is split into the following options:

- **EmitWarningIfMathHeaderIsIncluded**
Specifies whether a warning is emitted if the C standard library header `math.h` is included. Default is `off`.
- **UseTLImplementationForFabsFunction**
Specifies whether `fabs` function from the C standard library header `math.h` is implemented as a conditional expression using the `?` operator directly in the production code. Default is `off`.

In addition, the `t1_fabs` dsfxp file is not included in the production code, because the Code Generator calls no longer the macro defined there, but generates a conditional expression using the `?` operator.

Related documentation

- [New Code Generator Options](#) on page 194
- [Obsolete Code Generator Options](#) on page 247
- [Other Code Changes](#) on page 237

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 have been used since TargetLink 4.1.

If a model-based option value is identical to the old default value, it is automatically changed to the new default value during the upgrade. If a DD-based option value is identical to 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 earlier 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 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 is identical to 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 result in different option values. Refer to the following table.

If the default values for TargetLink versions A, B, and C are 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 is identical to the default.

New Code Generator options For more information on new Code Generator options, refer to [New Code Generator Options \(TargetLink New Features and Migration Guide !\[\]\(99f58673407353e96a019fbca558fd72_img.jpg\)](#)).

Related topics

References

- [EmitWarningIfMathHeaderIsIncluded \(TargetLink Model Element Reference !\[\]\(4e333a6106fc298d0ae6dff272a736ef_img.jpg\)](#))
- [UseTLImplementationForFabsFunction \(TargetLink Model Element Reference !\[\]\(97089f8e07e24e31baa67366e358a709_img.jpg\)](#))

API Functions and Hook Scripts

Where to go from here

Information in this section

Changes in API Functions.....	210
Changes in Hook Scripts.....	214

Changes in API Functions

Building and compiling production code

You can now build and compile production code directly from the Data Dictionary.

New properties For `t1_build_host` and `t1_build_target`:

Property	Description
CompilerOptions	Lets you add compiler flags. This only works for building from the Data Dictionary.
SimConfigName	Only for <code>t1_build_target</code> . Specifies the simulation configuration name. Default is <code>default EVM</code> .

For `tl_compile_host` and `tl_compile_target`:

Property	Description
DDModules	Specifies the list of DD Module or DD ModuleGroup objects which production is to be compiled for. If this property is used without the <code>Model</code> and <code>TLSubsystem</code> properties, the code is compiled from the Data Dictionary.
CompilerOptions	Lets you add compiler flags. This only works for building from the Data Dictionary.
SimConfigName	Only for <code>tl_compile_target</code> . Specifies the simulation configuration name. Default is <code>default EVM</code> .

Changed properties For `tl_build_host` and `tl_build_target`:

Property	Description
DDCodeGeneratorUnits	If this property is used without the <code>Model</code> and <code>TLSubsystem</code> properties, the code is generated and compiled from the Data Dictionary.
DDModules	If this property is used without the <code>Model</code> and <code>TLSubsystem</code> properties, the code is generated and compiled from the Data Dictionary.


For `tl_compile_host` and `tl_compile_target`:

Property	Description
DDCodeGeneratorUnits	If this property is used without the <code>Model</code> and <code>TLSubsystem</code> properties, the production code is compiled from the Data Dictionary.
DDModules	If this property is used without the <code>Model</code> and <code>TLSubsystem</code> properties, the production code is compiled from the Data Dictionary.

Removed properties For `tl_compile_host` and `tl_compile_target`:

- `RebuildAll`

Related documentation



- [tl_build_host](#) (TargetLink API Reference )
- [tl_build_target](#) (TargetLink API Reference )
- [tl_compile_host](#) (TargetLink API Reference )
- [tl_compile_target](#) (TargetLink API Reference )

dsdd_export_a2l_file

New properties

Property	Description
ASAP2Version	Specifies the ASAM MCD-2 MC standard version. Possible values are <code>1.61</code> and <code>1.71</code> . Default is <code>1.61</code> .

Related documentation

- [dsdd_export_a2l_file](#) (TargetLink API Reference )
- [Basics on the A2L File Format](#) (TargetLink Interoperation and Exchange Guide )

tl_clean**New properties**

Property	Description
bForceDelete	Specifies whether the files for the specified code generation units are deleted without asking for confirmation. Default is <code>false</code> .

Related documentation

- [tl_clean](#) (TargetLink API Reference )

tl_generate_fmu**New properties**

Property	Description
FmiVersion	Specifies FMI version. Possible values are <code>2.0</code> or <code>3.0</code> . Default is <code>3.0</code> .

Related documentation

- [tl_generate_fmu](#) (TargetLink API Reference )

tl_get

You can now retrieve the full DD path of an AUTOSAR DD element from an AUTOSAR block.

Related documentation

- [tl_get](#) (TargetLink API Reference )

tlEnumDataType**New properties**

Property	Purpose
Overwrite	Specifies whether to overwrite or merge (default) existing Simulink enumeration data type specification files.
SLEnumType	Specifies the name of the Simulink enumeration data type. This property is ignored if the Simulink enumeration data type name is already specified by a DD EnumTemplate object.
CreateTemplate	Creates a DD EnumTemplate object that maintains the mapping between the Simulink enumeration data type and the DD Typedef object. This property is ignored if the DD EnumTemplate object already exists.

Related documentation

- [tlEnumDataType \(TargetLink API Reference !\[\]\(2824aab9645d9fab95bae27ff6828dab_img.jpg\)\)](#)
- [Basics on Working with Enumeration Data Types in TargetLink \(TargetLink Preparation and Simulation Guide !\[\]\(0fbf3ad74a6c8dc44ba9ea17fc2aca5e_img.jpg\)\)](#)

tl_pref**New properties**

Property	Purpose
importtolinux	Imports the TLPreferences.xml to the linux platform.

Related documentation

- [tl_pref \(TargetLink API Reference !\[\]\(223f1a84e0bc2cacb9c165f716817dcc_img.jpg\)\)](#)

tl_generate_swc_model**Renamed properties**

Old Name	New Name
AddRunnableTriggerPort	AddTriggerPortToRunnableSubsystem
AddOperationSubsystemTriggerPort	AddTriggerPortToOperationSubsystem
BusObjectNamePrefix	SimulinkDataTypePrefix
UpdateBusObjectWithMappedDDTypedef	UpdateSimulinkDataTypeWithDDTypedef

The old names are still supported to enable downward compatibility. In future TargetLink versions, they will be removed.

Removed properties

- AutosarVersion
- ModelClientServerPorts
- AddOperationCallTriggerPort

Related documentation

- [tl_generate_swc_model \(TargetLink API Reference !\[\]\(5d60fe8e38bc12bfb78103fc624e324c_img.jpg\)\)](#)
- [Basics on Adaptive AUTOSAR Frame Models \(TargetLink Adaptive AUTOSAR Modeling Guide !\[\]\(ffcc3930f6e82d7cb586237ada9d3332_img.jpg\)\)](#)
- [Basics on AUTOSAR Development Approaches and Frame Model Generation in TargetLink \(TargetLink Classic AUTOSAR Modeling Guide !\[\]\(643201da8ca427135d452a5259d9e20e_img.jpg\)\)](#)
- [Basics on Simulink® Bus Objects Used in TargetLink \(TargetLink Preparation and Simulation Guide !\[\]\(4c625d6c32d16066b13eb52e34c8435a_img.jpg\)\)](#)

dsdd ('Set', ...)

The behavior of the `dsdd('Set', ...)` API is changed for custom properties. If you do not explicitly specify a type for an existing property, the type is now kept and no longer changed to the default type.

Related documentation

- Set
- Custom Properties

CreateModulesForServiceInterfaces

The `CreateModulesForServiceInterfaceTypes` API function is renamed to `CreateModulesForAdaptiveAutosarInterfaceTypes`. The appropriate DD property in `/Pool/Autosar/Config/ImportExport` using the `dsdd_master_adaptive_autosar.ddjson` [System] template is also renamed.

Related documentation


- [CreateModulesForAdaptiveAutosarInterfaceTypes](#)

Changes in Hook Scripts



tl_post_add_arportblock_hook

The hook script now also works with frame generation for Adaptive AUTOSAR. Therefore the following new predefined variables are added:

Predefined Variable	Description
<code>aarSubsystemPortInfo</code>	Structure describing the current port. Applies for Adaptive AUTOSAR only.

For more information about the structure component, refer to [tl_post_add_arportblock_hook](#) (TargetLink File Reference .

Related documentation

- [tl_post_add_arportblock_hook](#) (TargetLink File Reference )
- [Basics on Adaptive AUTOSAR Frame Models](#) (TargetLink Adaptive AUTOSAR Modeling Guide )

tl_post_add_comspecblock_hook

The hook script now also works with frame generation for Adaptive AUTOSAR.

The predefined variable `hRunnablePort` was renamed to `hConnectedBlock`. The predefined variable `hRunnablePort` is still supported for downward compatibility, but it will be removed in the future TargetLink releases.


Related documentation [tl_post_add_comspecblock_hook](#) (TargetLink File Reference )

tl_post_add_swcsystem_hook

The hook script now also works with frame generation for Adaptive AUTOSAR.


The following new predefined variables are added:

Predefined Variable	Description
<code>aarFunctionSubsystemInfoList</code>	Vector of <code>aarSubsystemInfo</code> structures describing all the adaptive AUTOSAR function and method behavior subsystems of the software component.



Related documentation [tl_post_add_swcsystem_hook](#) (TargetLink File Reference )

tl_pre_add_arportblock_hook The hook script now also works with frame generation for Adaptive AUTOSAR. Therefore the following new predefined variables are added:


Predefined Variable	Description
aarSubsystemPortInfo	Structure describing the current port. Applies for Adaptive AUTOSAR only.

For more information about the structure component, refer to [tl_pre_add_arportblock_hook](#) (TargetLink File Reference .

Related documentation

- [tl_pre_add_arportblock_hook](#) (TargetLink File Reference )
- [Basics on Adaptive AUTOSAR Frame Models](#) (TargetLink Adaptive AUTOSAR Modeling Guide )

tl_pre_add_comspecblock_hook The hook script now also works with frame generation for Adaptive AUTOSAR. The predefined variable `hRunnablePort` was renamed to `hConnectedBlock`. The predefined variable `hRunnablePort` is still supported for downward compatibility, but it will be removed in the future TargetLink releases.

Related documentation [tl_pre_add_comspecblock_hook](#) (TargetLink File Reference )

tl_pre_procode_sim_hook If an error occurs when using the hook function `tl_pre_procode_sim_hook`, TargetLink displays an error message and cancels simulation.

Related documentation

- [tl_pre_procode_sim_hook](#) (TargetLink File Reference )


Other Migration Aspects

Various Migration Aspects

Position of DD Function objects in the Subsystems area of the Data Dictionary

In the Subsystems area, the position of all DD Function objects of a module changes. With TargetLink 2023-B, they are encapsulated below a DD object named Functions. If you access these DD objects using a script, you have to adapt it according to the new position in the Subsystems area.

Related documentation

- [Basics on the Data Dictionary Structure \(DD Object Tree\)](#) (TargetLink Data Dictionary Basic Concepts Guide )

New file format of Data Dictionary predefined system templates

The file format of the Data Dictionary predefined system templates changed from `.dd` to `.ddjson`. Scripts that access predefined system templates by their original name must be adapted.

Related documentation

- [Basics on DD File Formats \(TargetLink Data Dictionary Basic Concepts Guide !\[\]\(48a7667d09d5a06397e047ee4537bb6f_img.jpg\)\)](#)
- [How to Create DD Workspaces \(TargetLink Data Dictionary Basic Concepts Guide !\[\]\(3df135a685d1b545c4fa64a5f3516545_img.jpg\)\)](#)

Avoid IF_* variables

With TargetLink 2023-B, more IF_* variables are generated in the non-optimized code. The following methods let you prevent this behavior:

Method	Instruction	Implication
Activate optimization	Select the Generate optimized code checkbox on the Advanced page of the TargetLink Main Dialog and set the InliningThreshold code generator option to ≥ 1 .	This method can result in code changes. Refer to Major or Breaking Changes for Atomic Subsystems and Function Interfaces on page 217.
Activate Ensure output is virtual	Select the Ensure output is virtual checkbox in the Simulink dialog of Outport blocks.	The MIL behavior can change. Refer to Simulink documentation by MathWorks® .

Related code changes [Additional IF_* variables in simplified initialization mode](#) on page 218

MATLAB Support

No Support of MATLAB R2023a and R2023b

Initially, TargetLink 2023-B (23.1) does not support MATLAB® R2023a and R2023b. Support of MATLAB R2023a and R2023b will be provided with TargetLink 23.1 Patch 1. For a download, refer to <https://www.dspace.com/go/PatchesTL>.

Code Changes Between TargetLink 2022-B and TargetLink 2023-B

Where to go from here

Information in this section

Major or Breaking Changes for Atomic Subsystems and Function Interfaces.....	217
Code Efficiency.....	222
Adaptive-AUTOSAR-Related and CPP-Related.....	230
Struct-Related and Array-of-Struct-Related.....	232
Classic-AUTOSAR-Related.....	235
Other Code Changes.....	237

Major or Breaking Changes for Atomic Subsystems and Function Interfaces

Introduction

Earlier TargetLink versions contained a bug that could result in incorrect code (MIL/SIL differences) in the following context:

- Outports of conditionally executed or iterated subsystems and the inheritance of initial values from predecessor blocks to the subsystem's output in simplified initialization mode.

With TargetLink 2023-B, this bug was fixed.

To fix this bug, the way in which TargetLink determines whether **IF_*** variables for the outport of the conditionally executed or iterated subsystem are necessary, had to be changed. This can result in additional **IF_*** variables in production code, even in contexts not directly affected by the bug.

To mitigate this change, additional changes regarding the creation of **IF_*** variables were introduced that allow for better removal of **IF_*** variables or structurally identical code with **IF_*** variables replacing other variables.

Note

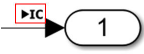
Your production code will be mostly unaffected if the following conditions hold:

- Your specifications do not prevent function inlining.
- You use DD VariableClass objects only if necessary for modeling.
- The model is configured to use classic initialization mode.

The support of classic initialization mode will be discontinued and removed in a future TargetLink version. Refer to [Features to Be Discontinued](#) on page 248.

Additional IF_* variables in simplified initialization mode

In simplified initialization mode, if a non-enhanced Outport block of a conditionally executed subsystem or iterated subsystem does not inherit an initial value from its predecessor block, TargetLink creates a variable for the output with the naming scheme *IF_<Name of outport>*. The following table shows the differences between previous TargetLink versions and the current one:

TargetLink < 2023-B	TargetLink 2023-B
No IF_* variable is created if it was specified at the outport that the initial value <i>is to be inherited</i> .	No IF_* variable is created if an initial value <i>is actually inherited</i> . In Simulink, actual inheritance is visualized by the IC icon on the signal line:  Also refer to Conditional Subsystem Initial Output Values in the Simulink documentation.

The additional IF_* variables can be eliminated by the optimization if the function resulting from the atomic subsystem in the initially generated code is inlined. The output variables of the predecessor blocks of the subsystem outputs are usually created with different initial scope and without initial value and the additional IF_* variable introduces additional degrees of freedom for TargetLink optimization with respect to the optimization order and matching optimization algorithms. Thus, compared to TargetLink < 2023-B (23.1), additional code changes as shown below can occur:

- Additional comments
- Variable renaming
- Missing variable initializations
 - Different order of variable declarations/definitions
 - Variables moved to control flows
- Different statement order
- For variables with access functions, initially generated cross-functional access patterns may become function-local access patterns or vice versa. Accordingly, the generated code might be more efficient, less efficient, or the communication paradigm at its function interface might change from communication via access functions to communication via the IF_* variable.

- For iterated subsystems, the Code Generator generates an output variable for the Simulink outputs if the preceding block is an Assignment block. This results in less efficient code that cannot be optimized.

Reason

- Bug fix
- Resolves differences in MIL/SIL/PIL simulation modes

Migration issue for conditionally executed subsystems If Simulink allows it, and if you can accept the change in MIL/model semantics, you can activate the Ensure output is virtual checkbox of the output of the conditionally executed subsystem.

Migration issue for iterated subsystems In many cases, you can activate the Ensure output is virtual checkbox of the output of the iterated subsystem without changing MIL/model semantics if all of the following conditions hold:

1. In its surrounding, the iterated subsystem is iterated at least once
2. The predecessor block of the Simulink output of the iterated subsystem always completely writes its output

If the predecessor block is an Assignment block, further actions might be required.

The code pattern of the Assignment block in its default configuration meets the second condition by initializing the complete block output Y by its input Y0.

If you changed the default behavior of the Assignment block by enabling the Omit dispensable initializations checkbox or specifying the variable class of the output variable to be MERGEABLE, it is no longer guaranteed that the block output output is completely written and the following conditions must apply:

- The Iterator block must drive the index port of the Assignment block.
- In its environment, the iterated system must iterate over all the elements of the output variable of the Assignment block.

Additional information Refer to [Impact and interaction](#) on page 221.

More structured IF_* variables

TargetLink 2023-B generates one structured IF_* variable instead of multiple non-structured IF_* variables in more situations.

In detail, an IF_* variable more often gets the structured type of the output variable of the predecessor block, if the following conditions hold:

- A bus signal is connected to a unenhanced Inport or Outport block whose predecessor block has a structured output variable in the production code.
- TargetLink must create an IF_* variable for this Inport or Outport block.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre> UInt32 IF_Sa2_Output = 0; UInt16 IF_Sa2_Output_a = 0; ... Float64 IF_Sa2_Output_g[6] = { /* [0..5] */ 0., 0., 0., 0., 0., 0. /* 0., 0., 0., 0., 0., 0. */ }; UInt32 IF_Sa2_Output_h = 0; struct e_new_Component_e_a_t_tag Sa5_Selector1; IF_Sa2_Output = Sa5_Selector1.e_a_a; IF_Sa2_Output_a = Sa5_Selector1.e_a_b.e_a_b_a.e_a_b_a_a; ... for (Aux_ = 0; Aux_ < 6; Aux_++) { IF_Sa2_Output_g[Aux_] = Sa5_Selector1.e_a_c.e_a_c_a[Aux_]; } IF_Sa2_Output_h = Sa5_Selector1.e_a_d; </pre>	<pre> struct e_new_Component_e_a_t_tag IF_Sa2_Output = { 0, /* e_a_a */ { 0, /* e_a_b_a_a */ ... } }, { { /* [0..5] */ 0., 0., 0., 0., 0., 0. /* 0., 0., 0., 0., 0., 0. */ } /* e_a_c_a */ }, 0 /* e_a_d */ }; struct e_new_Component_e_a_t_tag Sa5_Selector1; IF_Sa2_Output = Sa5_Selector1; </pre>

Reason

- Code efficiency
- Code size
- Easier matching of model and code
- Increased consistency
- Readability

Additional information Refer to [Impact and interaction](#) on page 221.

More consistent use of the variable class templates SLFunctionInputStruct and SLFunctionOutputStruct

For unenhanced Simulink ports, which are connected to bus signals, the variable class templates SLFunctionInputStruct and SLFunctionOutputStruct determine how to implement these ports in the production code. If a structured variable is connected to such a port, by default, it is implemented as call-by-reference function parameter with TargetLink 2023-B.

TargetLink ≤ 2022-B only implements such ports as function parameters if the predecessor block is implemented as call-by-reference using a struct type. If this was not the case, TargetLink generated global, non-structured IF_* variables for the appropriate ports.

The name template for an actual parameter does not start with IF_. If both TargetLink ≤ 2022-B and TargetLink 2023-B have structured IF_* variables and the function resulting from the atomic subsystem is inlined, then the IF_* variable seemingly loses its IF_ prefix.

Reason

- Bug fix
- Code efficiency
- Increased consistency
- MISRA C compliance
- Precision
- Readability
- Reduction of compile dependencies

Additional information Refer to [Impact and interaction](#) on page 221.

Impact and interaction

The mentioned code changes may interact in a way that leads to more complex code changes:

Function reuse and multiple instantiable AUTOSAR software components

- The change from global interface variables to call-by-reference communication effectively removes the respective variables from reuse or auto-created PIM structures unless an initial value is necessary for the actual parameter.

Refer to [More consistent use of the variable class templates SLFunctionInputStruct and SLFunctionOutputStruct](#) on page 220.

- The change from individual numerical to structured interface variables might lead to additional components in reuse structures because unnecessary components of the struct type that are tied up in struct assignments are not removed or replaced by other variables.

Refer to [More structured IF_* variables](#) on page 219.

- The loss of scope and initial value for predecessor variables changes reuse and auto-created PIM structures.

Refer to [Additional IF_* variables in simplified initialization mode](#) on page 218.

- In combination with changes in the sorting of struct components, the layout of these implicitly generated struct types may strongly differ from prior TargetLink versions.

Refer to [Changed sorting order of struct components of implicit structs](#) on page 233 and [Sorting of struct components for implicitly generated bus structs](#).

Code efficiency at function boundaries

- *Additional IF_* variables in simplified initialization mode* often affects code efficiency adversely, especially if function boundaries are not dissolved via inlining.

Refer to [Additional IF_* variables in simplified initialization mode](#) on page 218.

- *More structured IF_* variables* usually improves code efficiency for function boundaries that are not dissolved via inlining and compensates the adverse changes as long as predecessor variables can be eliminated.

Refer to [More structured IF_* variables](#) on page 219.

- *More consistent use of the variable class templates SLFunctionInputStruct and SLFunctionOutputStruct* usually decreases RAM consumption for function boundaries that are not dissolved via inlining at the cost of increased stack consumption and might prevent elimination and subsumption of preceding

states that are not always written (such as outputs of conditionally executed subsystems).

More consistent use of the variable class templates `SLFunctionInputStruct` and `SLFunctionOutputStruct` on page 220.

- See also Structured variables at Stateflow inputs.

Includes *More structured IF_* variables* and might cause `t1_basetypes` and `udt` header includes switching between header and source file.

Refer to [More structured IF_* variables](#) on page 219 and [More consistent use of the variable class templates SLFunctionInputStruct and SLFunctionOutputStruct](#) on page 220.

Related topics

References

[Assignment Block \(TargetLink Model Element Reference\)](#)

Code Efficiency

Improved elimination of array-of-struct intermediate variables in custom code

In the context of custom code, TargetLink can now eliminate array-of-struct intermediate variables, if the custom code input accesses a subset of the defined range:

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++) { A2[Aux_S32] = A1[Aux_S32]; } /* Custom code */ { Sa1_OutPort = A2[Sa1_InPort - 1].m12; }</pre>	<pre>/* Custom code */ { Sa1_OutPort = A1[Sa1_InPort - 1].m12; }</pre>

Reason

- Code efficiency

Improved elimination of intermediate variables in signal split situations

TargetLink's capability to replace temporary variables that result from signal splits was improved even if the value assignment takes place conditionally. Removal now is possible for struct variables and array-of-struct variables if the removed variable and the substituting variable have the same initial values.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>if (Sa1_InPort1 != 0) { LastValid_State = Sa1_InPort; } Sa1_OutPort1 = LastValid_State.c0; Sa1_OutPort = LastValid_State ;</pre>	<pre>if (Sa1_InPort1 != 0) { Sa1_OutPort = Sa1_InPort; } Sa1_OutPort1 = Sa1_OutPort.c0;</pre>

Additionally, numerical variables can be replaced if the symbol to be replaced has aliases, i.e., the symbol is a pointer or there are pointers to the symbol.

Reason

- Code efficiency
- Code size
- Code stability
- Matches user expectations
- Easier debugging
- Easier matching of model and code
- Facilitates compilability

Improved elimination of intermediate variables of the struct/array-of-struct type via code scheduling

TargetLink moves blocking statements in order to create a situation where elimination of struct intermediate variables is possible.

Example 1

TargetLink ≤ 2022-B
<pre>Sa4_In.RunA_NonAROut = NonAROut_RunA; Sa4_In.RootIn3 = InPort3_Root; NonARBusInport_RunB = X_Unit_Delay_Root; // blocks the replacement of Sa4_In by X_Unit_Delay_Root X_Unit_Delay_Root = Sa4_In;</pre>
TargetLink 2023-B
<pre>NonARBusInport_RunB = X_Unit_Delay_Root; X_Unit_Delay_Root.RunA_NonAROut = NonAROut_RunA; X_Unit_Delay_Root.RootIn3 = InPort3_Root;</pre>

Example 2

TargetLink ≤ 2022-B
<pre>Sa2_BusCreation(&BusCreationOut1, BusCreationOut2_signal1, &BusCreationOut2_signal3); for (Aux_S32 = 0; Aux_S32 < EW1; Aux_S32++) { Sa1_signal1_a[Aux_S32] = X_Sa1_Unit_Delay[Aux_S32]; X_Sa1_Unit_Delay[Aux_S32] = BusCreationOut2_signal1[Aux_S32]; Sa1_signal2_a[Aux_S32] = X_Sa1_Unit_Delay_a[Aux_S32]; X_Sa1_Unit_Delay_a[Aux_S32] = BusCreationOut2_signal2[Aux_S32]; } Sa1_signal3_a = X_Sa1_Unit_Delay_b; X_Sa1_Unit_Delay_b = BusCreationOut2_signal3; for (Aux_S32 = 0; Aux_S32 < 8; Aux_S32++) { Sa1_signal1[Aux_S32] = X_Sa1_Unit_Delay1_signal1[Aux_S32]; Sa1_signal2[Aux_S32] = X_Sa1_Unit_Delay1_signal2[Aux_S32]; } Sa1_signal3 = X_Sa1_Unit_Delay1_signal3; X_Sa1_Unit_Delay1 = BusCreationOut1;</pre>

TargetLink 2023-B

```

for (Aux_S32 = 0; Aux_S32 < EW1; Aux_S32++)
{
    Sa1_signal1_a[Aux_S32] = X_Sa1_Unit_Delay[Aux_S32];
    Sa1_signal2_a[Aux_S32] = X_Sa1_Unit_Delay_a[Aux_S32];
}

Sa1_signal3_a = X_Sa1_Unit_Delay_b;

for (Aux_S32 = 0; Aux_S32 < 8; Aux_S32++)
{
    Sa1_signal1[Aux_S32] = X_Sa1_Unit_Delay1.signal1[Aux_S32];
    Sa1_signal2[Aux_S32] = X_Sa1_Unit_Delay1.signal2[Aux_S32];
}

Sa1_signal3 = X_Sa1_Unit_Delay1.signal3;
Sa2_BusCreation(&X_Sa1_Unit_Delay1, X_Sa1_Unit_Delay, &X_Sa1_Unit_Delay_b);

for (Aux_S32 = 0; Aux_S32 < EW1; Aux_S32++)
{
    X_Sa1_Unit_Delay_a[Aux_S32] = BusCreationOut2_signal2[Aux_S32];
}

```

Reason

- Code efficiency

Elimination of array-of-struct/struct intermediate variables in a = a situations

In the context of loops, TargetLink can now eliminate more intermediate variables of the struct/array-of-struct type, if the elimination results in an `a = a` situation.

Code example:

```

for (Aux_S32 = 0; Aux_S32 < 4; Aux_S32++)
{
    Sa1_DSR_AosM2x3_T3[Aux_S32] = DSM_AosT3[1][2].a1[Aux_S32];

    DSM_AosT3[1][2].a1[Aux_S32] = Sa1_DSR_AosM2x3_T3[Aux_S32];
}

DSM_AosT3[0][2].a1[Index] = Sa1_DSR_AosM2x3_T3[Index];

```

Is optimized to:

```
DSM_AosT3[0][2].a1[Index] = DSM_AosT3[1][2].a1[Index];
```

Reason

- Code efficiency

Elimination of RTE API return pointers for array-of-struct

In prior TargetLink versions, RTE API return pointers could be removed only for numerical variables and struct variables. As of now, TargetLink can remove RTE API return pointers for vector-of-struct variables. Analogous to numerical matrix variables, array-of-matrix variables are not removed due to the cast.

TargetLink ≤ 2022-B

```
const myStruct2 * p_MyImplicitIRV_T1_3;

p_MyImplicitIRV_T1_3 = Rte_IrvIRead_Run_MyImplicitIRV_T1_3();

Sa3_Selector1 = p_MyImplicitIRV_T1_3[1];
```

TargetLink 2023-B

```
Sa3_Selector1 = Rte_IrvIRead_Run_MyImplicitIRV_T1_3()[1];
```

Reason

- Code efficiency
- Increased consistency

Elimination of intermediate variables for array-of-struct components

TargetLink can now eliminate struct components with non-scalar struct type as intermediate variables in situations where subsequently only one element is selected:

TargetLink ≤ 2022-B

```
for i
{
    S.v[i] = InAos[3].v[i]
}

Out = S.v[MyIdx]
```

TargetLink 2023-B

```
Out = InAos[3].v[MyIdx]
```

Reason

- Code efficiency
- Code size

Code duplication in conditionally executed control flows

With TargetLink 2023-B, the optimization of conditionally executed control flows was improved.

TargetLink now can move function calls consuming *output variables* of control flow statements such as Switch block outputs into mutually exclusive conditionally executed control flow branches by placing the respective function call in each control flow branch. The goal of this transformation is elimination of the former output variable inside at least part of the branches and reduced average stack consumption if no elimination is possible.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre> Int16 o; if (...) { o = q; } else { if (...) { funcDef1(o); } else { funcDef2(o); } } funcUse(o); </pre>	<pre> if (...) { funcUse(q); } else { if (...) { Int16 o_a; funcDef1(o_a); funcUse(o_a); } else { Int16 o_b; funcDef2(o_b); funcUse(o_b); } } </pre>

The removal of the output variable alone saves memory. The size of the code does not necessarily increase, because this optimization facilitates follow-up optimizations.

You control this optimization via the `UtilizeCodeDuplicationInControlFlowBranches` Code Generator option.

Also refer to [Write accesses to buffer variables in conditional control flow branches](#) on page 236.

Reason

- Code efficiency
- Code size
- Matches user expectations

Migration issue This optimization can conflict with the creation of local value copies of access functions. Refer to the `DD CreateLocalValueCopy` property.

Simplification of index-accesses for array-of-struct variables

TargetLink now simplifies the index accesses of array-of-struct variables on root or substruct level. In particular, casts are now removed and algebraic simplification and constant folding is performed.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre> AoS[(T) <expr>].c AoS[idx].sub[(T) (2 - 1)].c </pre>	<pre> AoS[<expr>].c AoS[idx].sub[1].c </pre>

Reason

- Code efficiency
- Matches user expectations
- Readability
- Suppress superfluous code

Initial value propagation for index expressions

TargetLink now propagates initial values of read-only variables that are exclusively located in index expressions. In prior versions, at least one access outside index accesses was necessary in order to trigger this optimization.

Starting Point	TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>T a = 5; T c = 3; b[a] = a + d[c];</pre>	<pre>T c = 3; b[5] = 5 + d[c];</pre>	<pre>b[5] = 5 + d[3];</pre>

Reason

- Bug fix
- Code efficiency
- Matches user expectations
- Increased consistency
- Side effect
- Suppress superfluous code

Structured variables at Stateflow inputs

Instead of specifying plain variables for Stateflow inputs, which are connected to bus signals and for which the Create input variable checkbox is cleared, the following structured variables are used or created:

- Structured predecessor variables
- Structured IF variables

Example 1

Note

The struct assignment contains the modeled copy of the entire struct, previously unused components can be omitted.

TargetLink ≤ 2022-B

```
for (Sc6_For_Iterator_it = 1; Sc6_For_Iterator_it <= Sc1_Dim1_1; Sc6_For_Iterator_it++)
{
...
for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++)
{
MyAoSType3_1_e_Assignment[Sc1_Dim1_1 - 1].y_2[Aux_S32] =
MyAoSType3_1_c[1].m_1.z_3[Sc6_For_Iterator_it - 1].y_2[Aux_S32];
}

MyAoSType3_1_e_Assignment[Sc1_Dim1_1 - 1].y_3 =
MyAoSType3_1_c[1].m_1.z_3[Sc6_For_Iterator_it - 1].y_3;
...
}
```

TargetLink 2023-B

```
for (Sc6_For_Iterator_it = 1; Sc6_For_Iterator_it <= Sc1_Dim1_1; Sc6_For_Iterator_it++)
{
...
MyAoSType3_1_e_Assignment[Sc1_Dim1_1 - 1] =
MyAoSType3_1_c[1].m_1.z_3[Sc6_For_Iterator_it - 1];
...
}
```

Example 2

TargetLink ≤ 2022-B

```

static Float64 Ca1_inbus;
static Float64 Ca1_inbus_a;
static Int32 Ca1_inbus_b[5];
static Int32 Ca1_inbus_b[5];
static Int8 Ca1_inbus_c;
static UInt8 Ca1_inbus_e[5];

static void Ca1_Chart1(void)
{
    Int32 Aux_S32;

    Ca1_outbus.y_3.v_1 = Ca1_inbus_c;
    Ca1_outbus.y_3.v_2 = Ca1_inbus_d;
    for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++)

    {
        Ca1_outbus.y_2[Aux_S32] = Ca1_inbus_b[Aux_S32];
        Ca1_outbus.y_3.v_3[Aux_S32][0] = Ca1_inbus_e[Aux_S32];
    }
    Ca1_outbus.y_3.v_4[0] = Ca1_inbus_f[0];
    Ca1_outbus.y_3.v_4[1] = Ca1_inbus_f[1];
    Ca1_outbus.y_1.x_1.w = Ca1_inbus + 0.01;
    Ca1_outbus.y_1.x_2 = Ca1_inbus_a + 0.02;
}

static void Sa6_ForIterator_AoS3_1_a(...)
{
    ...
    for (Sa6_For_Iterator_it = 1;
        Sa6_For_Iterator_it <= ((Int16) Sa6_Dimension1);
        Sa6_For_Iterator_it++)
    {
        ...
        Ca1_inbus = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_1.x_1.w;
        Ca1_inbus_a = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_1.x_2;
        Ca1_inbus_c = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_1;
        Ca1_inbus_d = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_2;

        for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++)
        {
            Ca1_inbus_b[Aux_S32] = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_2[Aux_S32];
            Ca1_inbus_e[Aux_S32] = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_3[Aux_S32][0];
        }
        Ca1_inbus_f[0] = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_4[0];
        Ca1_inbus_f[1] = MyAoSType3_1_m_1__6_.z_3[Sa6_For_Iterator_it - 1].y_3.v_4[1];

        Ca1_Chart1();
    }
    ...
}

```

TargetLink 2023-B

```

static struct m_1_z_3_t_tag Sa6_Selector1;

static void Ca1_Chart1(void)
{
    Ca1_outbus = Sa6_Selector1;

    Ca1_outbus.y_1.x_1.w = Sa6_Selector1.y_1.x_1.w + 0.01;
    Ca1_outbus.y_1.x_2 = Sa6_Selector1.y_1.x_2 + 0.02;
}

static void Sa6_ForIterator_AoST3_1_a(...)
{
    ...
    for (Sa6_For_Iterator_it = 1;
        Sa6_For_Iterator_it <= ((Int16) Sa6_Dimension1);
        Sa6_For_Iterator_it++)
    {
        ...
        Sa6_Selector1 =
            MyAoSType3_1_m_1_6_z_3[Sa6_For_Iterator_it - 1];

        Ca1_Chart1();
        ...
    }
    ...
}

```

Reason

- Bug fix
- Code efficiency
- Code size
- Increased consistency
- Readability

Migration issue In situations similar to Example 1 with only one or only few components of the Stateflow chart input accessed inside the chart, struct copy can be less efficient for struct types with many and/or large components.

You can use the following workarounds:

- Use a Bus Selector block, select the relevant bus elements, and adjust the Stateflow chart input from struct to the respective components' number and data type.
- Adjust the `StructAssignmentDecompositionAuxVarMemoryUsageThreshold` Code Generator option or `StructAssignmentDecompositionStatementMultiplicationThreshold` Code Generator option to allow TargetLink to break up the struct assignment and get rid of either the unnecessary components or the unnecessary assignment to unused parts of the struct variable.

Related topics

References

[CreateLocalValueCopy \(TargetLink Data Dictionary Reference !\[\]\(eafc244b53721dd1ec133f0772f70fc7_img.jpg\)\)](#)
[Switch Block \(TargetLink Model Element Reference !\[\]\(cb741e910ae1fce3b15fcd4605753ff5_img.jpg\)\)](#)
[UtilizeCodeDuplicationInControlFlowBranches \(TargetLink Model Element Reference !\[\]\(7db78e01f48713b9a2242a4e52c8494a_img.jpg\)\)](#)

Adaptive-AUTOSAR-Related and CPP-Related

Cpp casts in cpp files

Cpp files now generally use C++ casts instead of C casts.

C++ Casts	Description
<code>static_cast<T>(expr)</code>	Replaces most numeric C casts.
<code>reinterpret_cast<T>(expr)</code>	Leads to a reinterpretation of the bit pattern. This C++ cast is used with AddressMapped macros and row pointer casts.
<code>const_cast<T>(expr)</code>	This C++ cast is used to change the <code>const</code> or <code>volatile</code> property of a pointer target.

Reason

- C++ Standard Compliance

Changed namespace qualification

The new Code Generator option `NamespaceQualification` has changed the namespace qualification of symbols in the generated C++ code. By default, the Code Generator uses a fully qualified namespace prefixed with `::`. This prevents accidental matching of identifiers from identically named namespaces, e.g., `A::x` matching `::A::x` and `::A::B::A::x` if inside `::A::B`. The namespaces `std` and `ara` are excluded from this.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>namespace A { namespace B { using MyType = std::int32_t; A::B::MyType MyVar = 5; struct MyStruct { MyType a; }; } void func() { A::B::MyType MyLocalVar = A::B::MyVar; } }</pre>	<pre>namespace A { namespace B { using MyType = std::int32_t; ::A::B::MyType MyVar = 5; struct MyStruct { ::A::B::MyType a; }; } void func() { ::A::B::MyType MyLocalVar = ::A::B::MyVar; } }</pre>

Reason

- Bug fix
- Precision

Opening curly bracket at namespace declarations

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>namespace A { namespace B { using MyType = std::int32_t; } }</pre>	<pre>namespace A { namespace B { using MyType = std::int32_t; } }</pre>

Reason

- Increased consistency
- Readability

Changes in enumeration data types implemented as macros

For enumeration data types that are implemented as macros and whose CreateTypedef property is set to off, no namespaces are declared and the closing brace of a namespace is no longer output on the same line as the last enum element.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>namespace MyNamespace { using EnumWithNamespace_CreateTypeDefOn = std::int32_t; #define Red3 (static_cast<EnumWithNamespace_CreateTypeDefOn>(1)) #define Yellow3 (static_cast<EnumWithNamespace_CreateTypeDefOn>(0)) #define Red (static_cast<std::int32_t>(1)) #define Yellow (static_cast<std::int32_t>(0)) } }</pre>	<pre>#define Red (static_cast<std::int32_t>(1)) #define Yellow (static_cast<std::int32_t>(0)) namespace MyNamespace { using EnumWithNamespace_CreateTypeDefOn = std::int32_t; #define Red3 (static_cast<EnumWithNamespace_CreateTypeDefOn>(1)) #define Yellow3 (static_cast<EnumWithNamespace_CreateTypeDefOn>(0)) }</pre>

Reason

- Bug fix
- Readability

Type guard names contain namespaces

For data types with constraints, type guards are created in the code. If these data types also have a C++ namespace, these types are given an extended type guard name that includes the individual namespace names, e.g., <NS1>_<NS2>_...<NSn>_<Typename>_TYPE.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>namespace my { namespace types { #ifndef MyInt16_WithNS_TYPE #define MyInt16_WithNS_TYPE using MyInt16_WithNS = std::int16_t; #endif /* MyInt16_WithNS_TYPE */ } }</pre>	<pre>namespace my { namespace types { #ifndef my_types_MyInt16_WithNS_TYPE #define my_types_MyInt16_WithNS_TYPE using MyInt16_WithNS = std::int16_t; #endif /* my_types_MyInt16_WithNS_TYPE */ } }</pre>

Reason

- Code stability
- Readability
- C++ Standard Compliance

Language comment in file header

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>*** Simulink model : aar_communication *** TargetLink subsystem : aar_communication/S1 *** Codefile : S1.h *** *** Generated by TargetLink, the dSPACE production quality code generator</pre>	<pre>*** Simulink model : aar_communication *** TargetLink subsystem : aar_communication/S1 *** Codefile : S1.h *** Language : Cpp *** *** Generated by TargetLink, the dSPACE production quality code generator</pre>

Reason

- Matches user expectations

Struct-Related and Array-of-Struct-Related

In more modeling situations bus-capable blocks inherit structured variables

A bus-capable block whose Inherit Properties checkbox is selected is connected to a bus signal or an array-of-bus signal. This block tries to inherit a structured data type from its predecessor block to generate an output variable or a block parameter with a structured data type or even an array-of-struct data type. This is not always possible. In such a case, the following happens:

- If the block tries to generate a variable with an array-of-struct type, the code generation cancels with an error.
- If the block tries to generate one structured variable, n non-structured variables are created instead.

With TargetLink 2023-B, there are more situations where such a block can inherit a structured data type from its predecessor block:

- The structured DD Typedef object has a component that references a DD Typedef object. This DD Typedef object has constraints and its CreateTypedef property is set to **off**.

- The Min/Max values for a leaf struct component of the data type are set.

Reason

- Bug fix
- Code efficiency
- Code size
- Increased consistency

Consistent handling of struct variables

Unused components of an implicit struct variable are created and synchronized in the same way as instance components of a fixed data type.

In addition, refer to the following code changes:

- The **Original size...** comment is only used for n-D tables of reuse structures.
- The **Description, Min,** and **Max** values of the unused implicit struct variable components are not shown.

Reason

- Bug fix
- Increased consistency

Changed sorting order of struct components of implicit structs

The sorting criteria for struct components changed:

Sorting Criterium	Change
BaseType	TargetLink now first sorts by the name of the basetype and then by the number of dimensions.
Type	TargetLink now first sorts by the name of the type and then by the number of dimensions.
Size	TargetLink now includes the dimensionality of struct components into the size calculation. For example, a components of type <code>Int16[2]</code> now counts as 32-bit instead of 16-bit .

This affects implicitly generated structs in the following contexts:

- Function reuse
- Variant coding
- Implicit structs generated from bus port blocks

Reason

- Bug fix
- Code efficiency
- Matches user expectations

Migration issue If you need a fixed struct type layout, then use a struct DD Typedef object without `COMPONENTS_SORTABLE` in its Optimization property.

Additional information Refer to [Impact and interaction](#) on page 221.

Sorting of struct components for implicitly generated bus structs

When sorting the struct components of structs that are generated from bus port blocks whose Create implicit struct checkbox is selected, TargetLink now takes substructs into account. This can result in a different order of struct components and in different names of the substruct types:

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre> struct BSS_IP_Sa2_InPort { Int32 InPort_b_i32; Int16 Sa2_a_i16; }; struct BSS_IP_Sa2_InPort_a { Int32 InPort_b_i32; UInt32 Sa2_InPort; Int16 InPort_a_i16; }; struct BSS_IP_Sa2_InPort_b { Int32 Sa2_InPort_a[3]; Int16 Sa2_InPort; }; /* Description: bus inport struct */ struct BS_IP_Sa2_InPort { Int32 InPort_f_i32_vec5[5]; Int32 InPort_b_i32; Int16 InPort_d_i16_vec5[5]; Int16 Sa2_a_i16; struct BSS_IP_Sa2_InPort InPort_e_sub; struct BSS_IP_Sa2_InPort_a InPort_l_sub2; struct BSS_IP_Sa2_InPort_b InPort_l_sub3; }; </pre>	<pre> struct BSS_IP_Sa2_InPort { Int32 Sa2_InPort_a[3]; Int16 Sa2_InPort; }; struct BSS_IP_Sa2_InPort_a { Int32 InPort_b_i32; UInt32 Sa2_InPort; Int16 InPort_a_i16; }; struct BSS_IP_Sa2_InPort_b { Int32 InPort_b_i32; Int16 Sa2_a_i16; }; struct BS_IP_Sa2_InPort { struct BSS_IP_Sa2_InPort InPort_l_sub3; struct BSS_IP_Sa2_InPort_a InPort_l_sub2; struct BSS_IP_Sa2_InPort_b InPort_e_sub; Int32 InPort_f_i32_vec5[5]; Int32 InPort_b_i32; Int16 InPort_d_i16_vec5[5]; Int16 Sa2_a_i16; }; </pre>

Reason

- Bug fix
- Code efficiency
- Matches user expectations
- Increased consistency

Additional information Refer to [Impact and interaction](#) on page 221.

User-defined variable classes of unused structure components

User-defined variable classes of unused structure components are now considered, regardless of whether the structure type is explicit or implicit.

This might result to the following code changes:

- For the unused component, an access function is generated.
- If the Optimization property of the user-defined variable class is set to ERASABLE, the unused component is omitted from the data type.

Reason

- Bug fix
- Matches user expectations
- Increased consistency

Reduce singleton dimensions

In the following situations, the dimension is always reduced:

- Array-of-Struct
- Only for the root dimension of a struct.
- No DD Variable object
- [1x1] matrix and [1] vector
- Dimension is not explicitly specified.

Reason

- Bug fix

Sorting of temporary variables in restart functions

Temporary variables that are generated to initialize an array of struct variable get the variable class `SLLocalInit` instead of `SLLocal`. As a result, the sorting of variable declarations in restart functions changes.

TargetLink ≤ 2022-B

```
/* SLLocal: Default storage class for local variables | Width: N.A. */
ShV5hV5h_tp M2x2_ShV5hV5h_Aux = {
    0, /* Comp1Scalar */
    ...
};

/* SLLocal: Default storage class for local variables | Width: 32 */
Int32 Aux_S32;
```

TargetLink 2023-B

```
/* SLLocal: Default storage class for local variables | Width: 32 */
Int32 Aux_S32;

/* SLLocalInit: Default storage class for local variables with init value | Width: N.A. */
ShV5hV5h_tp M2x2_ShV5hV5h_Aux = {
    0, /* Comp1Scalar */
    ...
};
```

Reason

- Bug fix
- Increased consistency

Classic-AUTOSAR-Related

Data type of Classic AUTOSAR Arrays

The data type of Classic AUTOSAR arrays has changed: they are no longer named by the underlying TargetLink base type. Instead, the type name of the scalar element is used.

Consider a DD Typedef object called `MyArrayType` with the following settings:

- `CreateTypedef` set to `off`
- `BaseType` set to `sint8`
- `AutosarArrayElementRef` set to `MyScalarType`

The code changed as follows:

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>sint8 DE_Array[3]; Rte_Read_Swc_R_Port_SRI_DE_Array(DE_Array);</pre>	<pre>MyScalarType DE_Array[3]; Rte_Read_Swc_R_Port_SRI_DE_Array(DE_Array);</pre>

This also affects formal parameters of server runnables:

```
void Runnable(MyScalarType * Arg_Matrix)
{
    MyScalarType (*p_Arg_Matrix)[2];
    p_Arg_Matrix = (sint8 (*)[2]) Arg_Matrix;
```

Casts are not affected because these always result from base types.

Reason

- Bug fix
- Matches user expectations
- Increased consistency

Write accesses to buffer variables in conditional control flow branches

The code generated for the following RTE API functions might have changed:

- Rte_Write
- Rte_IWrite
- Rte_IrvWrite
- Rte_IrvIWrite

Write accesses that pass complete buffer variables are now done in parallel in conditional control flows (`if ... then ... else` constructs).

Refer to [Code duplication in conditionally executed control flows](#) on page 225.

Superfluous cast to the operation result type for AUTOSAR arrays is omitted

TargetLink no longer generates a cast to the operation result type for AUTOSAR arrays if two 32-bit operands are added or subtracted.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>I32Out = (Int32)(I32VecIn1[0] + I32VecIn2[0]);</pre>	<pre>I32Out = I32VecIn1[0] + I32VecIn2[0];</pre>

Reason

- Bug fix
- Code efficiency
- Increased consistency
- Readability
- Suppress superfluous code

Other Code Changes

Macros in Stateflow code

When generating code from Stateflow charts, a macro with a user-defined variable class is replaced with its initial value, if the user-defined variable classes has the Optimization property set to ERASABLE and the OmitCastOfMacroValue property to on.

Reason

- Bug fix
- Code efficiency
- Increased consistency

Improved SQRT pattern

The SQRT pattern has been improved when the input to an SQRT block is an integer and the result is floating-point.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>if(in < 0) { out = 0; } else { out = sqrt(in); }</pre>	<pre>if(in <= 0) { out = 0; } else { out = sqrt(in); }</pre>

Reason

- Code efficiency

Integer casts for enumerations in LOG macro calls

Enumerations are represented as integers in the internal logging structure. Therefore, the Code Generator now includes casts to the underlying integer type of the enumerations in the call to the LOG macro.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>LOG_MATRIX(MyEnumMatrix, _MyEnumMatrix, 3, 2, MyEnumMatrix);</pre>	<pre>LOG_MATRIX(MyEnumMatrix, _MyEnumMatrix, 3, 2, (Int32) MyEnumMatrix);</pre>

Reason

- Bug fix

Signedness on enumeration casts

The signedness on enumeration cast has been fixed. In previous TargetLink versions enumeration types may have claimed to be signed, even if their underlying enumeration data type was unsigned.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>Aux__b = (Int32) (((Int32) Ca1_LightColorsIn) << 1);</pre>	<pre>Aux__b = (Int32) (((UInt32) Ca1_LightColorsIn) << 1);</pre>

Reason

- Bug fix

Superfluous `tl_basetypes.h` includes If a module only contains of enumeration data types, the `tl_basetypes.h` includes are removed.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>#include "tl_basetypes.h" typedef enum ... enum T..._tag ...</pre>	<pre>typedef enum ... enum T..._tag ...</pre>

- Reason**
- Bug fix
 - Matches user expectations

Order within statement comments at function calls and duplicate comment parts The order of comment parts for the actual parameters of a function call has changed:

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>/* call of function: Subsystem/F64FcnU640Outside # combined # update(s) for inport Subsystem/F64FcnU640Outside/In3 # combined # update(s) for inport Subsystem/F64FcnU640Outside/In2 # combined # update(s) for inport Subsystem/F64FcnU640Outside/In1 */ Sa2_F64FcnU640Outside((Float64) Sa1_U64Source, 1, 0., &Sa2_Out1);</pre>	<pre>/* call of function: Subsystem/F64FcnU640Outside # combined # update(s) for inport Subsystem/F64FcnU640Outside/In1 # combined # update(s) for inport Subsystem/F64FcnU640Outside/In2 # combined # update(s) for inport Subsystem/F64FcnU640Outside/In3 */ Sa2_F64FcnU640Outside((Float64) Sa1_U64Source, 1, 0., &Sa2_Out1);</pre>

Changes for `# combined #` comment:

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>/* Sum: Subsystem/Sum # combined # Unit delay: Subsystem/Unit Delay # combined # TargetLink output: Subsystem/Out1 # combined # Unit delay: Subsystem/Unit Delay */ Sa1_Out1 = (Int16) (Sa1_InPort + Unit_Delay_2);</pre>	<pre>/* Sum: Subsystem/Sum # combined # Unit delay: Subsystem/Unit Delay # combined # TargetLink output: Subsystem/Out1 */ Sa1_Out1 = (Int16) (Sa1_InPort + Unit_Delay_2);</pre>
<pre>/* # combined # Custom code: Path/to/Custom Code Block: << output code >> Update of input(s) for custom code block : Path/to/Custom Code Block */ { Sa2_Custom_Code_Block_y = (((uint16) 1)) << (Var1); }</pre>	<pre>/* Custom code: Path/to/Custom Code Block: << output code >> Update of input(s) for custom code block : Path/to/Custom Code Block */ { Sa2_Custom_Code_Block_y = (((uint16) 1)) << (Var1); }</pre>
<pre>/* Path1/to/Block1: Omitted comparison with constant. update(s) for inport Path1/to/Block2 Path1/to/Block3: Omitted comparison with constant. # combined # update(s) for inport Path1/to/Block4 # combined # update(s) for inport Path1/to/Block5 */ if (Ca4_Var1 && Var2 && Ca5_Var3) {?</pre>	<pre>/* Path1/to/Block1: Omitted comparison with constant. # combined # update(s) for inport Path1/to/Block2 Path1/to/Block3: Omitted comparison with constant. # combined # update(s) for inport Path1/to/Block4 # combined # update(s) for inport Path1/to/Block5 */ if (Ca4_Var1 && Var2 && Ca5_Var3) {??</pre>
<pre>/* Switch: Path1/to/SwitchBlock1 # combined # Product: Path2/to/ProductBlock1 # combined # Unit delay: Path3/to/UnitDelayBlock1 */ Aux_S16 += (X_Sa2_Unit_Delay * Sa3_Abs1);</pre>	<pre>/* # combined # Switch: Path1/to/SwitchBlock1 # combined # Product: Path2/to/ProductBlock1 # combined # Unit delay: Path3/to/UnitDelayBlock1 */ Aux_S16 += (X_Sa2_Unit_Delay * Sa3_Abs1);</pre>

Reason

- Bug fix
- Increased consistency
- Readability

Indentation of function declarations and definitions in preprocessor control flow

As for variable declarations and definitions in preprocessor control flow, the function declarations and definitions are also indented.

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>#if U extern void Sa2_ss_A(void); #endif</pre>	<pre>#if U extern void Sa2_ss_A(void); #endif</pre>

Reason

- Increased consistency

Volatile is omitted when reducing to auto

The `volatile` keyword is excluded from a variable when it is reduced to automatic storage duration. This triggers the output of Advice 31419. However, if the variable has a variable class with scope reduction chain defined via `ScopeReducedClass` entry, then the `Volatile` property of the variable class for `Scope=local`, `Storage=default` specifies the desired behavior and no advice is given.

Reason

- Bug fix
- Code efficiency
- Suppress superfluous code

BlockType in comments

For most blocks, the `BlockType` is set in the comments of the production code as follows:

- The mask type without the `TL_` prefix.
- The Simulink block type

Reason

- Bug fix
- Increased consistency
- Readability

Limit macros in MinMax block pattern

If a `MinMax` block has a single non-scalar input, TargetLink generates the following code for the max function:

```
MinMaxBlockOut = MinValueOfBlockOutputType;
if(In[0] > MinMaxBlockOut) MinMaxBlockOut = In[0];
...
if(In[n] > MinMaxBlockOut) MinMaxBlockOut = In[n];
```

For floating-point, the generated code will be as follows:

TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>MinMaxBlockOut = -1.7976931348623157e+308; if ...</pre>	<pre>MinMaxBlockOut = t1_FLOAT64NMIN; if ...</pre>
The changes also apply for Float32 and the min function.	

Reason

- Bug fix
- Code stability
- Increased consistency
- Readability

ConformingFreestanding fabs

If you activate the UseTLImplementationForFabsFunction Code Generator option, formerly known as ConformingFreestandingImplementation in prior versions, TargetLink uses a conditional expression.

TargetLink ≤ 2022-B
<pre>#include "t1_fabs.h" Sa1_AbsF64Out = fabs(Sa1_F64In);</pre>
TargetLink 2023-B
<pre>Sa1_AbsF64Out = (Sa1_F64In < 0.) ? (-Sa1_F64In) : Sa1_F64In;</pre>

Reason

- MISRA C compliance

Migration issue Refer to [ConformingFreestandingImplementation](#) on page 208.

Optimization order change can lead to code differences for relations

TargetLink's internal optimization order has been adapted in order to reduce code generation times. In the following special situation involving a zero-based For Iterator subsystem, the generated code is slightly changed.

Starting Point	TargetLink ≤ 2022-B	TargetLink 2023-B
<pre>DimSize = 3; Aux_ = (Int16) (DimSize - 1); for (it = 0; it <= Aux_; it++) { ... }</pre>	<pre>for (it = 0; it <= (Int16) 2; it++) { ... }</pre>	<pre>for (it = 0; it < (Int16) 3; it++) { ... }</pre>

Reason

- Bug fix
- Side effect

Int types are using the long form with signed for all targets

The type headers generated from `TargetConfig.xml` using the long form with signed.

```
typedef signed short int t1_dsfxp_Int16; /* Description: signed 16 bit integer */
typedef signed long int t1_dsfxp_Int32; /* Description: signed 32 bit integer */
typedef signed long long int t1_dsfxp_Int64; /* Description: signed 64 bit integer */
typedef signed char t1_dsfxp_Int8; /* Description: signed 8 bit integer */
```

Reason

- Bug fix
- MISRA C compliance

Additional optimizations for vector or matrix variables

If there are accesses to vector or matrix variables that include a fixed subrange of the variable with iteration for at least one dimension, additional optimizations may occur. The following example shows a column-wise rescheduling of state updates that can facilitate subsequent optimizations in extended situations:

TargetLink ≤ 2022-B

```
for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++)
{
    /* TargetLink Outport: TL_Root/IRead_Inport_Matrix/Out1 */
    Formal[Aux_S32][0] = pISV->X_SReuse1_Unit_Delay[Aux_S32][0];
    Formal[Aux_S32][1] = pISV->X_SReuse1_Unit_Delay[Aux_S32][1];

    /* UnitDelay: TL_Root/IRead_Inport_Matrix/Unit Delay */
    pISV->X_SReuse1_Unit_Delay[Aux_S32][0] = p_Matrix5x3[Aux_S32][0];
    pISV->X_SReuse1_Unit_Delay[Aux_S32][1] = p_Matrix5x3[Aux_S32][1];
}
```

TargetLink 2023-B

```
for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++)
{
    /* TargetLink Outport: TL_Root/IRead_Inport_Matrix/Out1 */
    Formal[Aux_S32][0] = pISV->X_SReuse1_Unit_Delay[Aux_S32][0];

    /* UnitDelay: TL_Root/IRead_Inport_Matrix/Unit Delay */
    pISV->X_SReuse1_Unit_Delay[Aux_S32][0] = p_Matrix5x3[Aux_S32][0];

    /* TargetLink Outport: TL_Root/IRead_Inport_Matrix/Out1 */
    Formal[Aux_S32][1] = pISV->X_SReuse1_Unit_Delay[Aux_S32][1];

    /* UnitDelay: TL_Root/IRead_Inport_Matrix/Unit Delay */
    pISV->X_SReuse1_Unit_Delay[Aux_S32][1] = p_Matrix5x3[Aux_S32][1];
}
```

Reason

- Code efficiency
- Code size

Omission of unnecessary pointer-to-const casts

With the support of C++ casts, const casts are now omitted, especially in Classic AUTOSAR projects. In order to support certain target compilers, TargetLink historically inserted casts in order to add type qualifiers to the first level of

indirection (usually pointer to const) even though this is unnecessary in standard C.

TargetLink now uses C++ cast types and does not insert these unnecessary casts. For structural equality between C and C++ code, these casts also are omitted in C code. This change mostly affects Classic AUTOSAR projects and Lookup Tables (function call or map struct initialization).

For example:

- ```
void myFcn(const Int16 * PointerToArray);
```

```
Int16 MyArray[5];
```
- Function reuse

| TargetLink ≤ 2022-B                               | TargetLink 2023-B                  |
|---------------------------------------------------|------------------------------------|
| <pre>myFcn((const Int16 *)&amp;MyArray[0]);</pre> | <pre>myFcn(&amp;MyArray[0]);</pre> |

**Pointer-init-cast omitted** In older TargetLink versions, a cast was generated when initializing pointers with the address of a variable if the two types differed in a type prefix. This is no longer the case with TargetLink 2023-B.

In the following example, the map structure has the type prefix `MyQual`. The `z_table` also has this type prefix, but the component to which the address of `z_table` is assigned does not have the type prefix.

| TargetLink ≤ 2022-B                                                                                                                                                                                                              | TargetLink 2023-B                                                                                                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>MyQual MAP_Tab2DS0I2T16676_a Look_Up_Table__2D__map = {     2, /* Nx: Description: number of axis#1 points */     ...     (const Int16 *) &amp;(MyTable[0][0]) /* z_table:     Description: matrix table values */ };</pre> | <pre>MyQual MAP_Tab2DS0I2T16676_a Look_Up_Table__2D__map = {     2, /* Nx: Description: number of axis#1 points */     ...     &amp;(MyTable[0][0]) /* z_table: Description: matrix table     values */ };</pre> |

**Reason**

- Suppress superfluous code
- Readability
- MISRA C++ Guidelines
- Increased consistency

**Using TargetLink basetypes in casts**

Casts to TargetLink basetypes, e.g., Int16 are now applied more consistently.

| TargetLink ≤ 2022-B                                                                | TargetLink 2023-B                                                              |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| <pre>SetAbsAlarm(MyAlarm, (TickType) TL_FX_GROUND, (TickType) TL_FX_GROUND);</pre> | <pre>SetAbsAlarm(MyAlarm, (UInt32) TL_FX_GROUND, (UInt32) TL_FX_GROUND);</pre> |

Blocks that compute a scalar product with more than two inputs can change the cast:

| TargetLink ≤ 2022-B                                                                                                                                     | TargetLink 2023-B                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>X_Sa11_DiscrFilter[0] += ((THETA) (((THETA) (((Int32) Sa11_Input) + ((Int32) X_Sa11_DiscrFilter[1])) &gt;&gt; 1)) - X_Sa11_DiscrFilter[0]));</pre> | <pre>X_Sa11_DiscrFilter[0] += ((Int16) (((Int16) (((Int32) Sa11_Input) + ((Int32) X_Sa11_DiscrFilter[1])) &gt;&gt; 1)) - X_Sa11_DiscrFilter[0]));</pre> |

**Reason**

- Bug fix
- Increased consistency

**Migration issue**

**Changed optimization for external subsystems and subsystems configured for incremental code generation**

TargetLink cannot know the contents of external subsystems and subsystems configured for incremental code generation, and assumes that they are unknown or variable. This means that TargetLink cannot determine whether the output variables are completely described each time they are called.

As of TargetLink 2023-B, TargetLink considers output parameters of external subsystems and subsystems configured for incremental code generation as conditionally or incompletely described during optimization unless the step function class referenced from the Function block contains `SIDE_EFFECT_FREE` in its Optimization property. As a result, optimizations that affect the output of these subsystems may no longer be executed.

**Reason**

- Bug fix
- Increased consistency

**Removed include from `tlnomath.h`**

TargetLink provides its own implementation in the `Dsfxp` library for calculating the math function `hypot()`, which is not part of the standard C library. This implementation calls the `sqrt()` function and therefore includes `math.h`. For this reason, the `tlnomath.h` include is removed.

| TargetLink ≤ 2022-B                                          | TargetLink 2023-B                   |
|--------------------------------------------------------------|-------------------------------------|
| <pre>#include "tlnomath.h"... Out = C_HYPOT(In1, In2);</pre> | <pre>Out = C_HYPOT(In1, In2);</pre> |

**Reason**

- Bug fix
- Side effect

**Improved variable names and order for 3-D, 4-D, 5-D look-up function calls**

The naming template for the implicit variables `Offset`, `Fraction`, and `LinIdx` in 3-D, 4-D, and 5-D look-up function calls has been changed to `$S_$B_X`. The preparations of these parameters of the look-up function call are grouped and a comment that indicates the model path to the block. For example:

```
/* Interpolation: Path/To/Interpolation Using Prelookup */
<Offset vector Assignments>
<Fraction vector Assignments>
<Computation of LinIdx>
```

In addition, descriptive comments are added for the generated 3-D, 4-D, and 5-D look-up function as follows:

- For table accesses: `Gather all values`
- For interpolations: `Interpolation along dimension X`.

**Reason**

- AUTOSAR compliance
- Bug fix
- Matches user expectations
- Increased consistency
- Readability

**Changed identifiers for ap\_Entry and ap\_Exit calls**

TargetLink automatically assigns the identifiers for the `ap_Entry` and `ap_Exit` calls depending on the calling function, the nesting depth in the control flow, and the inlining order. Due to the solution of a problem that made the identifiers for the same code generation non-deterministic across TargetLink versions, swapped identifiers may occur in individual cases.

| TargetLink ≤ 2022-B                                                                                                                                                                                                                                                          | TargetLink 2023-B                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> if (Cond1) {   ap_Entry_SID1_Name_1_101();   ...   ap_GetVector_In(Aux_Vector_In_ID);   ...   ap_Exit_SID1_Name_1_101(); } ...  if (Cond2) {   ap_Entry_SID1_Name_1_100();   ...   ap_GetScalar_In(&amp;Aux_Scalar_In_ID);   ...   ap_Exit_SID1_Name_1_100(); } </pre> | <pre> if (Cond1) {   ap_Entry_SID1_Name_1_100();   ...   ap_GetVector_In(Aux_Vector_In_ID);   ...   ap_Exit_SID1_Name_1_100(); } ...  if (Cond2) {   ap_Entry_SID1_Name_1_101();   ...   ap_GetScalar_In(&amp;Aux_Scalar_In_ID);   ...   ap_Exit_SID1_Name_1_101(); } </pre> |

**Reason**

- Bug fix
- Code stability

**Loop merging with different loop variable types**

Loop merging with different loop variable types has been changed as follows:

- If a loop variable is replaced by a loop variable of a different type, casts may be inserted before the accesses of the loop variable.
- Loop variables with an address access are no longer swapped with loop variables of other types. This might result in less loop merging and other loop variables.
- Loop variables are not exchanged by loop variables of other signedness.

**Reason**

- Bug fix

**Blank line between enum macro definition and enum definition**

After the last macro definition for an enum with macro implementation, a blank line is now output if it is followed by a multiline statement, e.g., enum type definition.

| TargetLink ≤ 2022-B                                                                                                                                                                                                                                                                                                | TargetLink 2023-B                                                                                                                                                                                                                                                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>typedef Int32 MyBasicColors2; #define Black ((MyBasicColors2) 0) #define Pink ((MyBasicColors2) 1) #define Brown ((MyBasicColors2) 2) #define Magenta ((MyBasicColors2) 3) typedef enum MyBasicColors1_tag {     Red = 0,     Yellow = 1,     Green = 2,     Blue = 3,     Orange = 4 } MyBasicColors1;</pre> | <pre>typedef Int32 MyBasicColors2; #define Black ((MyBasicColors2) 0) #define Pink ((MyBasicColors2) 1) #define Brown ((MyBasicColors2) 2) #define Magenta ((MyBasicColors2) 3)  typedef enum MyBasicColors1_tag {     Red = 0,     Yellow = 1,     Green = 2,     Blue = 3,     Orange = 4 } MyBasicColors1;</pre> |

**Reason**

- Readability

**Comments in preprocessor statements**

Comments in preprocessor statements are now wrapped without backslash and like normal C comments.

| TargetLink ≤ 2022-B                                                                                                           | TargetLink 2023-B                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <pre>#define SetVarS16(_var, _value) (_var = _value) /* Use SetVarTp(&lt;Name&gt;) macro to write a scalar\ variable */</pre> | <pre>#define SetVarS16(_var, _value) (_var = _value) /* Use SetVarTp(&lt;Name&gt;) macro to write a scalar vari able */</pre> |

**Reason**

- Bug fix

**Related topics**

**References**

[Function Block \(TargetLink Model Element Reference\)](#)

## Discontinuations as of TargetLink 2023-B

**Where to go from here**

**Information in this section**

|                                       |     |
|---------------------------------------|-----|
| Discontinued TargetLink Features..... | 246 |
| Obsolete Limitations.....             | 246 |
| Obsolete Code Generator Options.....  | 247 |

## Discontinued TargetLink Features

### Discontinuation of the Microsoft Express Edition compiler

The Microsoft Express Edition compiler is no longer supported. Instead, you can use all editions (Community, Professional, and Enterprise) as of 2015 of the Microsoft Visual Studio C++ (MSVC) product family as MEX compiler and for debugging in SIL simulation mode. Since debugging in SIL simulation mode is only possible with the previously mentioned compilers, the `tlProductionCodeSILCompiler` API command is no longer required. For details on SIL debugging, refer to [How to Debug in SIL Simulation Mode \(TargetLink Preparation and Simulation Guide\)](#).

## Obsolete Limitations

### No const qualifier for function input

TargetLink does not support `const` qualification for function inputs that are implemented as function parameters, i.e., that are specified in one of the following ways:

- TargetLink InPort blocks with variables that reference DD VariableClass objects whose Scope property is set to `value_param` or `ref_param`.
- TargetLink Bus InPort blocks whose variables are specified as explicit structs and reference DD VariableClass objects whose Scope property is set to `ref_param`.

Const qualification of block output variables is possible only in the following cases:

- For the output variable of the TargetLink Constant block
- For TargetLink InPorts that are specified for indirect reuse, i.e., the InPort block references a DD VariableClass object that is specified as follows:

| Property                     | Value                         |
|------------------------------|-------------------------------|
| Scope                        | <code>ref_param</code>        |
| FunctionReuse                | <code>InstanceSpecific</code> |
| FunctionReuseAccessByPointer | <code>On</code>               |

### Restrictions concerning Adaptive AUTOSAR and code generation

TargetLink does not support code generation for models with a subsystem whose Function block has its Role set to `Adaptive AUTOSAR Function` if both of the following conditions apply:

- The subsystem is modeled directly below the TargetLink subsystem.
- The DD `AdaptiveAutosarFunction` object that is associated with the subsystem references a DD `Module` object that is influenced by a DD `ModuleTemplate` object.

### Pointer as return value

Calling a function with a return value of pointer type is not supported.

**Inconsistent behavior of the ConformingFreestandingImplementation option**

Activation of this option avoids standard `math.h` functions in the generated code, and a TargetLink-specific implementation is used instead. However, a TargetLink-specific implementation exists only for the `fabs` function.

## Obsolete Code Generator Options

| Removed Code Generator Option        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ConformingFreestandingImplementation | <p>The Code Generator option is split into the following new Code Generator options:</p> <ul style="list-style-type: none"> <li>▪ <b>EmitWarningIfMathHeaderIsIncluded</b><br/>Specifies whether a warning is emitted if the C standard library header <code>math.h</code> is included.</li> <li>▪ <b>UseTLImplementationForFabsFunction</b><br/>Specifies whether <code>fabs</code> function from the C standard library header <code>math.h</code> is implemented as a conditional expression using the <code>?</code> operator directly in the production code.</li> </ul> |

**Related topics****References**

[EmitWarningIfMathHeaderIsIncluded \(TargetLink Model Element Reference !\[\]\(642aa997563f9a325b310230bb5078b7\_img.jpg\)\)](#)  
[UseTLImplementationForFabsFunction \(TargetLink Model Element Reference !\[\]\(9bef82f5a53106f2ad06a2de7acf5bcf\_img.jpg\)\)](#)

# Changes in Future TargetLink Versions

## Where to go from here

## Information in this section

|                                        |     |
|----------------------------------------|-----|
| Features to Be Discontinued.....       | 248 |
| API Functions to Be Discontinued.....  | 249 |
| Deprecated Code Generator Options..... | 249 |

## Features to Be Discontinued

|                                                                                 |                                                                                                                                                             |
|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RTOS/OSEK code generation mode</b>                                           | Support for the TargetLink RTOS/OSEK code generation modes will be discontinued in a future TargetLink version.                                             |
| <b>Target Optimizations Module (TOM)</b>                                        | Support for the TargetLink Target Optimizations Module (TOM) will be discontinued in a future TargetLink version.                                           |
| <b>Simulink classic initialization mode</b>                                     | Support for the Simulink classic initialization mode will be discontinued in a future TargetLink version.                                                   |
| <b>Dynamic components</b>                                                       | Support for specifying dynamic components for DD Variable objects will be discontinued in a future TargetLink version.                                      |
| <b>SWC ReceiverPort and SWC SenderPort blocks (TargetLink AUTOSAR Blockset)</b> | Support for SWC ReceiverPort and SWC SenderPort blocks will be discontinued in a future TargetLink version. Instead, InPort and OutPort blocks can be used. |

## Related topics

## References

- [InPort Block \(TargetLink Model Element Reference !\[\]\(4f6d8a8b127300a02d56d34d01423d15\_img.jpg\)\)](#)
- [OutPort Block \(TargetLink Model Element Reference !\[\]\(7e3d1ad67bf2d7a17700a66d1a313f91\_img.jpg\)\)](#)
- [SWC ReceiverPort Block \(TargetLink Model Element Reference !\[\]\(6aaf22e5a325c32ef2122c2939c64f9c\_img.jpg\)\)](#)
- [SWC SenderPort Block \(TargetLink Model Element Reference !\[\]\(27421322d686d4980ae2b7f101ee89ba\_img.jpg\)\)](#)



## API Functions to Be Discontinued

### Discontinued API functions








The following API functions are deprecated and will be removed in a future TargetLink version:

| Function                 | Deprecated Since | Replacement Function |
|--------------------------|------------------|----------------------|
| t1_compare_fcn_signature | TargetLink 5.0   | -                    |

## 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 )
- [DeriveDefaultSymbolClasses](#) (TargetLink Model Element Reference )
- [ExploitRangesIndependentOfErasable](#) (TargetLink Model Element Reference )
- [HandleUnscaledStateflowExpressionsWithTIType](#) (TargetLink Model Element Reference )
- [StrictTypedefPlacement](#) (TargetLink Model Element Reference )
- [UseRootFileNameAsUserDefinedTypesHeaderBasename](#) (TargetLink Model Element Reference )
- [UtilizeValueEqualityStructFunctionArguments](#) (TargetLink Model Element Reference )



# VEOS

## Where to go from here

## Information in this section

|                                                                                                |     |
|------------------------------------------------------------------------------------------------|-----|
| <a href="#">New Features of VEOS 2023-B</a> .....                                              | 251 |
| Provides an overview of the new features of VEOS 2023-B.                                       |     |
| <a href="#">Compatibility of VEOS 2023-B</a> .....                                             | 253 |
| Provides information on the compatibility of VEOS 2023-B.                                      |     |
| <a href="#">Migrating to VEOS 2023-B</a> .....                                                 | 257 |
| To migrate from VEOS 2023-A to VEOS 2023-B, you might have to perform certain migration steps. |     |
| <a href="#">Discontinuations Planned for VEOS 2024-A</a> .....                                 | 258 |
| Provides information on the discontinuations planned for VEOS 2024-A.                          |     |

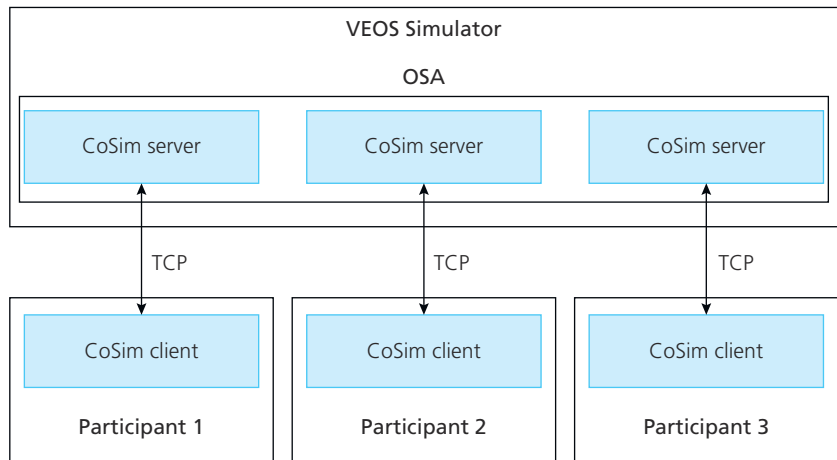
## New Features of VEOS 2023-B

### Co-simulation with VEOS

dSPACE provides a new interface for implementing time-synchronized co-simulations with VEOS. This lets you couple VEOS with domain-specific simulators, such as data replay tools and sensor signal simulators. The co-simulation interface enables the exchange of simulated CAN, LIN, and Ethernet bus messages as well as I/O signals.

Co-simulation with VEOS is implemented in a client-server architecture, where the VEOS Simulator is the co-simulation server, i.e., it provides the overall simulation time for all co-simulation participants. The interface is implemented as a generic C library.

Refer to the following illustration.



This feature requires the adaptive V-ECU license (refer to [VEOS Modules \(VEOS Introduction and Overview\)](#))).

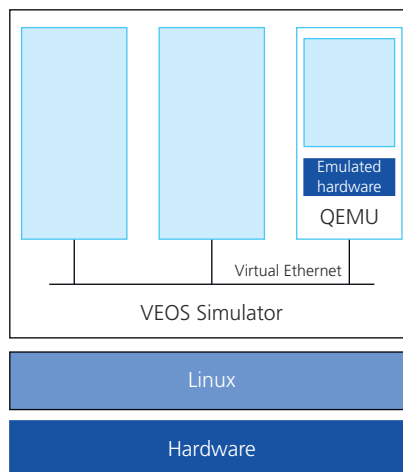
For more information, refer to [Basics on Co-Simulation with VEOS \(VEOS Manual\)](#))).

### QEMU integration into VEOS simulation

You can now integrate QEMU system emulation in a VEOS simulation. The emulation of ARM and x86 hardware architectures is supported, which allows you to run binaries that are not built for VEOS.

During QEMU system emulation, VEOS controls the QEMU virtual time. QEMU integration is currently supported for VEOS on Linux.

Refer to the following illustration:



This feature requires the adaptive V-ECU license (refer to [VEOS Modules \(VEOS Introduction and Overview\)](#))).

For more information, refer to [Basics on QEMU Integration in VEOS \(VEOS Manual\)](#))).

### VEOS Simulator Console enhancements

There are some enhancements for the VEOS Simulator Console:

- **info** command:  
The command now displays the elapsed real time and the acceleration factor.
- **start** and **step** commands:
  - You can now specify the simulation duration. For example:  
`veos-sim start 30s`: The simulation runs for 30 s after the `start` command is called. The duration refers to the virtual simulation time, not to the PC time.
  - In addition, the new `--blocking` option lets you specify that `veos-sim` does not return until the specified simulation duration has elapsed. For example:  
`veos-sim step 20s --blocking`: The simulation runs step-wise for 20 s after the `step` command is called. Then the VEOS Simulator Console returns.

Refer to [VEOS Simulator Console Command Reference \(VEOS Manual !\[\]\(e2376d476d06eb31946dc01a69a4403a\_img.jpg\)](#)).

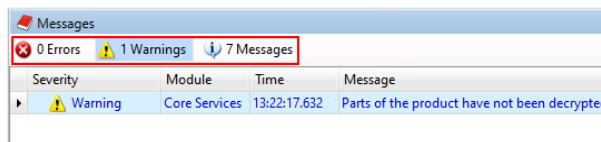
### Dev Container file for rebuilding the VEOS Linux kernel

There is now a Dev Container configuration file that lets you rebuild the VEOS Linux kernel for the GPU passthrough demo by simply using the Dev Containers extension of Visual Studio Code.

Refer to [How to Prepare the Host PC for Running the Fluid Simulation \(VEOS Manual !\[\]\(830769b31eeeaca920791081939ff8ba\_img.jpg\)](#)).

### VEOS Player: Filter settings of the Messages pane are auto-saved

The filter settings in the Messages (refer to [Messages Pane \(VEOS Manual !\[\]\(8bba887393ca45b761e5cb49e755e762\_img.jpg\)](#))) pane of the VEOS Player are now auto-saved and reapplied the next time you start the VEOS Player.



## Compatibility of VEOS 2023-B

### General compatibility

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 Integrating the Simulation System !\[\]\(5eb1325dfdc3f1cad8426726c0db51cd\_img.jpg\)\)](#).

**Note**

When you build model containers *in separate steps*, e.g., when you compile an existing model container in advance with the dSPACE Model Container Utility, dSPACE recommends you to use *compatible versions of the compiler and linker*. This ensures binary compatibility of the build result.

For more information, refer to [Ensuring Binary Compatibility When Building Model Containers \(VEOS Manual !\[\]\(950a62bbddad88d64435fd35607dfc42\_img.jpg\)\)](#).

**Supported operating systems**

VEOS is available for Windows and Linux. For information on the supported distributions, refer to [Operating System](#) on page 261.

**VEOS on Linux** For information on VEOS on Linux, refer to [Introduction to VEOS on Linux \(VEOS Manual !\[\]\(e1d6102fe77919492c04879c8450f1f5\_img.jpg\)\)](#).

**Virtualizing VEOS** For information on virtualizing VEOS for cloud/cluster environments, refer to [Virtualization of VEOS \(VEOS Manual !\[\]\(73002692dd5e7a64e60946be3158e719\_img.jpg\)\)](#).

**BSC file compatibility**

The following table shows the compatibility between VEOS 2023-B and bus simulation container (BSC) files:

| BSC Files Created With ...                                                                                                                                                                   | BSC Version |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <ul style="list-style-type: none"> <li>▪ Bus Manager 2023-B</li> <li>▪ Bus Manager 2023-A</li> </ul>                                                                                         | 1.12        |
| <ul style="list-style-type: none"> <li>▪ Bus Manager 2022-B</li> <li>▪ Ethernet Configuration Package 4.5<sup>1)</sup></li> <li>▪ Ethernet Configuration Package 4.4<sup>1)</sup></li> </ul> | 1.11        |

<sup>1)</sup> For detailed and up-to-date compatibility information regarding the Ethernet Configuration Package, refer to <http://www.dspace.com/go/ethcpinfo>.

- If a BSC file was generated *with an SIC file*, the target architecture selected for SIC file generation specifies the simulation targets available to the VEOS build process.

Refer to [Basics on Simulink Implementation Containers \(Model Interface Package for Simulink - Modeling Guide !\[\]\(b538fe54c1f3a7343e37e85cc2d00497\_img.jpg\)\)](#).

- If a BSC file was generated *without an SIC file*, you can select any simulation target supported by the Simulation Target Manager.

**FMU file compatibility**

VEOS supports Functional Mock-up Units (FMUs) that comply with the following FMI standard versions:

- FMI 2.0
- FMI 3.0

For more information on the FMI 3.0 release, refer to <https://fmi-standard.org/news/2022/05/10/fmi-3.0-release.html>.

For detailed and up-to-date compatibility information on dSPACE FMI support, refer to:

<http://www.dspace.com/go/FMI-Compatibility>.

### OSA file compatibility

The following table shows the compatibility between VEOS 2023-B and offline simulation application (OSA) files:

| OSA Files Created With ...       | OSA Version        |
|----------------------------------|--------------------|
| VEOS 2023-B                      | 23.2               |
| VEOS 2023-A                      | 23.1 <sup>1)</sup> |
| VEOS 2022-B                      | 22.2 <sup>1)</sup> |
| VEOS 5.4 (dSPACE Release 2022-A) | 5.4 <sup>1)</sup>  |

<sup>1)</sup> You cannot modify the properties of VPUs contained in an OSA file if you open the OSA file in a later VEOS version than the version with which the OSA file was originally created. However, you can edit port and network connections and enable/disable experimentation services. It is recommended that you rebuild the binary OSA files from existing model implementation container files (BSC, FMU, SIC, VECU) when you migrate from one VEOS version to another.

### SIC file compatibility

The following table shows the compatibility between VEOS 2023-B and Simulink implementation container (SIC) files:

| SIC Files Created With ...                                                                                                                                                          | SIC Version <sup>1)</sup> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| <ul style="list-style-type: none"> <li>▪ Model Interface Package for Simulink 2023-B</li> <li>▪ Model Interface Package for Simulink 2023-A</li> <li>▪ TargetLink 2023-B</li> </ul> | 1.12                      |
| <ul style="list-style-type: none"> <li>▪ Model Interface Package for Simulink 2022-B</li> <li>▪ TargetLink 2022-B</li> </ul>                                                        | 1.11 <sup>2)</sup>        |
| <ul style="list-style-type: none"> <li>▪ Model Interface Package for Simulink 22.1</li> <li>▪ Model Interface Package for Simulink 4.6</li> <li>▪ TargetLink 5.2</li> </ul>         | 1.10 <sup>2)</sup>        |

<sup>1)</sup> The target architecture selected for SIC file generation specifies the simulation targets available to the VEOS build process. Refer to [Basics on Simulink Implementation Containers \(Model Interface Package for Simulink - Modeling Guide\)](#).

<sup>2)</sup> 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 2023-B (dSPACE Release 2023-B). For more information, refer to [Migrating ASM Models \(VEOS New Features and Migration\)](#).

### SMC file compatibility

The following table shows the compatibility between VEOS 2023-B and system model container (SMC) files:

| SMC Files Created With ...                                                                                                                                | SMC Version |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <ul style="list-style-type: none"> <li>▪ VEOS 2023-B</li> <li>▪ VEOS 2023-A</li> <li>▪ VEOS 2022-B</li> <li>▪ VEOS 5.4 (dSPACE Release 2022-A)</li> </ul> | 1.2         |

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 2023-B imports neither the unsupported container nor the connections to the application process based on the unsupported container.

### VECU file compatibility

The following table shows the compatibility between VEOS 2023-B and V-ECU implementation container (VECU) files:

| V-ECU Implementations Created With...                                                                                                        | V-ECU Implementation Version |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| <ul style="list-style-type: none"> <li>▪ SystemDesk 2023-B</li> <li>▪ TargetLink 2023-B</li> </ul>                                           | 3.3                          |
| <ul style="list-style-type: none"> <li>▪ SystemDesk 2022-B</li> <li>▪ TargetLink 2022-B</li> </ul>                                           | 3.2                          |
| <ul style="list-style-type: none"> <li>▪ SystemDesk 5.6 (dSPACE Release 2021-B)</li> <li>▪ TargetLink 5.2 (dSPACE Release 2021-B)</li> </ul> | 3.1                          |

### Compatibility with Real-Time Testing

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     | RTT Version                      |
|--------------------|----------------------------------|
| ... of VEOS 2023-B | Real-Time Testing Version 2023-B |
| ... of VEOS 2023-A | Real-Time Testing Version 2023-A |
| ... of VEOS 2022-B | Real-Time Testing Version 2022-B |
| ... of VEOS 5.4    | Real-Time Testing Version 5.2    |
| ... of VEOS 5.3    | Real-Time Testing Version 5.1    |
| ... of VEOS 5.2    | Real-Time Testing Version 5.0    |
| ... of VEOS 5.1    | Real-Time Testing Version 4.4    |
| ... of VEOS 5.0    | Real-Time Testing Version 4.3    |

ControlDesk 2023-B automatically uses the VEOS Simulator of VEOS 2023-B. You can therefore use RTT in connection with VEOS and ControlDesk if Real-Time Testing Version 2023-A is active on the PC.

### Compatibility with Python

The automation interface of the VEOS Player in VEOS 2023-B is compatible with Python 3.11.



**Related topics****Basics**

[Ensuring Binary Compatibility When Building Model Containers \(VEOS Manual !\[\]\(666e09182d4cd268646ea700ea60dcdf\_img.jpg\)\)](#)  
[Hypervisor Configuration for Adaptive V-ECU Simulation on Windows \(VEOS Working with Adaptive V-ECUs !\[\]\(1ef1ef0bf9af6c6996401964cf280f2d\_img.jpg\)\)](#)

## Migrating to VEOS 2023-B

**Introduction**

To migrate from VEOS 2023-A to VEOS 2023-B, you might have to perform certain migration steps.

**Note**

To migrate to VEOS 2023-B from versions earlier than 2023-A, you might also have to perform the migration steps of the intervening VEOS versions.

**Migrating to Python 3.11**

As of dSPACE Release 2023-B, dSPACE products no longer support Python 3.9. Instead, Python 3.11 is supported.

There are no VEOS-specific migration aspects.

For more information, refer to [Migrating Python Scripts from Python 3.9 to Python 3.11](#) on page 29.

**Changed port names in SIC containers**

Model Interface Package for Simulink 2023-B provides enhanced support for a number of Simulink® port types. When you generate a Simulink implementation container (SIC) file, the naming of the root-level Simulink ports in the SIC file differs from the names generated by earlier Model Interface Package for Simulink versions.

As a result, if your offline simulation application contains a VPU based on a Simulink implementation container (SIC) file created with Model Interface Package for Simulink 2023-A or earlier and you recreate the SIC file with Model Interface Package for Simulink 2023-B or later and reimport it to VEOS, you get a warning message during import, and port connections of the related VPU are lost.

You have to reconnect these ports in VEOS.

For more information, refer to [Migrating to the Model Interface Package for Simulink 2023-B](#) on page 142.

**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 [Basics on Integrating Simulink Implementation Containers \(SICs\) \(VEOS Integrating the Simulation System\)](#).

---

### Migrating from earlier VEOS versions

To migrate from earlier 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 Planned for VEOS 2024-A

---

### No support for cross-compilation

As of dSPACE Release 2024-A, offline simulation applications for VEOS on Linux can no longer be built on Windows. This will apply to the VEOS Player and the VEOS Build Console.

To build offline simulation applications for VEOS on Linux, use the VEOS Build Console on Linux instead. Refer to [How to Build an Offline Simulation Application \(VEOS Integrating the Simulation System\)](#).

### No adaptive V-ECU simulation on Windows via Microsoft Hyper-V

As of dSPACE Release 2024-A, adaptive V-ECU simulation on Windows via Microsoft Hyper-V is no longer supported.

As an alternative, you can run the adaptive V-ECU in a QEMU emulation, and integrate the QEMU emulation in a VEOS simulation.

For more information, refer to [Basics on QEMU Integration in VEOS \(VEOS Integrating QEMU in VEOS\)](#).

# Compatibility Information

## Where to go from here

## Information in this section

|                                                      |     |
|------------------------------------------------------|-----|
| Supported MATLAB Releases.....                       | 259 |
| Operating System.....                                | 261 |
| Using dSPACE Software on Virtual Machines (VMs)..... | 263 |
| Run-Time Compatibility of dSPACE Software.....       | 268 |
| Limitations for Using Windows Features.....          | 268 |
| Limitations for Using Linux Features.....            | 270 |

## Supported MATLAB Releases

### MATLAB®/Simulink®

Various dSPACE products require a MATLAB installation on the host PC.

#### Tip

For system requirements of MathWorks® software, refer to <https://www.mathworks.com/support/sysreq.html>.

| MATLAB Release... | ...Is Supported by dSPACE Release 2023-B |                                        |                                     |                   |                      |                                               |                            |
|-------------------|------------------------------------------|----------------------------------------|-------------------------------------|-------------------|----------------------|-----------------------------------------------|----------------------------|
|                   | RCP and HIL 2023-B <sup>1), 2)</sup>     | ConfigurationDesk 2023-B <sup>3)</sup> | AutomationDesk 2023-B <sup>4)</sup> | TargetLink 2023-B | Model Compare 2023-B | dSPACE Python Extensions 2023-B <sup>5)</sup> | XIL API .NET MAPort 2023-B |
| R2023b            | ✓ <sup>6)</sup>                          | ✓                                      | ✓                                   | — <sup>7)</sup>   | ✓                    | ✓                                             | ✓                          |
| R2023a            | ✓ <sup>6)</sup>                          | ✓                                      | ✓                                   | — <sup>7)</sup>   | ✓                    | ✓                                             | ✓                          |
| R2022b            | ✓ <sup>6)</sup>                          | ✓                                      | ✓                                   | ✓                 | ✓                    | ✓                                             | ✓                          |
| R2022a            | ✓                                        | ✓                                      | ✓                                   | ✓                 | ✓                    | ✓                                             | ✓                          |

1) 'RCP and HIL software' is a generic term for a software package containing several dSPACE software products, for example, ASM, Model Interface Package for Simulink, RTI, and ModelDesk. These software products are installed in a common folder. As of dSPACE Release 2023-A, ConfigurationDesk is no longer part of the RCP and HIL software package. It is installed in its own folder.

2) MATLAB/Simulink Student Suite is not supported by Automotive Simulation Models (ASM).

3) ConfigurationDesk requires Model Interface Package for Simulink (part of the RCP and HIL software package) for a connection to MATLAB and to work with Simulink models. Thus, the RCP and HIL software installation defines the MATLAB support for ConfigurationDesk.

4) The AutomationDesk MATLAB Access Library requires MATLAB.

5) matlplib2 of dSPACE Python Extensions requires MATLAB.

6) R2023b, R2023a, and R2022b are not supported by the FPGA Programming Blockset 23.2 – FPGA Interface. For a workaround, refer to <http://www.dspace.com/faq?295>.

7) No support. Full support after installing the TargetLink 23.1 Patch 1. For a download, refer to <https://www.dspace.com/go/PatchesTL>.

For up-to-date information on additional MATLAB releases that can be used in combination with dSPACE software, refer to <https://www.dspace.com/go/MATLABCompatibility>.

## Operating System

---

### Windows operating system on host PC

The dSPACE products of dSPACE Release 2023-B support the following operating systems:

- The following editions, channels, and servicing options of Windows 10:
  - Windows 10 Professional, Education, and Enterprise (64-bit versions)  
The Windows 10 Home and Windows 10 S editions are not supported.
  - Long-Term Servicing Channel: LTSC 2019 and LTSC 2021
  - General Availability Channel (replaces the previous Semi-Annual Channel):  
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 22H2 version of the General Availability Channel for testing.
- The following editions, channels, and servicing options of Windows 11:
  - Windows 11 Professional, Education, and Enterprise  
The Windows 11 Home edition is not supported.
  - General Availability Channel: 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 22H2 version of the General Availability Channel for testing.

#### Note

While testing our products under Windows 11 22H2, a few issues occurred whose cause and remedy are not the responsibility of dSPACE. For details, refer to <https://www.dspace.com/go/Win11GAC22H2>.

- Windows Server LTSC 2019 Standard and Datacenter editions, each with the Desktop Experience installation option. Only the listed editions are supported. The Windows Server 2019 Essentials edition is not supported.
- Windows Server LTSC 2022 Standard and Datacenter editions, each with the Desktop Experience installation option. Only the listed editions are supported. The Windows Server 2022 Essentials edition is not supported.

Some limitations apply when you use dSPACE software in conjunction with features of Windows. Refer to [Limitations for Using Windows Features](#) on page 268.

**Support of Windows Docker** You can also run some dSPACE products in a Windows Docker container. For more details, contact dSPACE Support (<https://www.dspace.com/go/supportrequest>).

---

### Linux operating system on host PC

The dSPACE products of dSPACE Release 2023-B with Linux compatibility support the following operating system:

- Ubuntu 20.04 LTS with the General Availability Kernel in the Desktop, Server, and Cloud version each based on the AMD 64-bit processor architecture.
- Ubuntu 18.04 LTS is no longer supported.

Some limitations apply when you use dSPACE software in conjunction with features of Linux. Refer to [Limitations for Using Linux Features](#) on page 270.

**Support of Linux Docker** You can also run the dSPACE products with Linux compatibility in a Linux Docker container. For more details, contact dSPACE Support (<https://www.dspace.com/go/supportrequest>).

### 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. Any PC with CodeMeter Runtime software can be used as a license server.

**Valid for servers without dSPACE software** You can use operating systems from Microsoft Windows on a license server in combination with protected dSPACE software.

#### Note

- If you want to use Ubuntu Linux as operating system for the license server, contact dSPACE Support ([www.dspace.com/go/supportrequest](http://www.dspace.com/go/supportrequest)).
- Other operating systems 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>\dSPACE MotionDesk 2023-
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:
  - For Windows: 111 (TCP and UDP), 3702 (UDP), 7214 (TCP and TCP6), 7215 (TCP and UDP), 7216 (TCP), 8090 (TCP), 9923 (UDP), 49152 ... 65535 (TCP, TCP6 and UDP)

- For Linux: 111 (TCP and UDP), 7215 (TCP and UDP), 7216 (TCP), 8090 (TCP), 9923 (UDP), 32768 ... 60999 (TCP, TCP6 and UDP)
- MotionDesk requires the following open TCP/IP network port: 15000 (UDP)
- ConfigurationDesk used together with one or more connected MATLAB installations requires up to 10 open TCP/IP network ports in the range 49196 ... 49205 (TCP):
  - Whenever a connected MATLAB is started, the first available port from the specified range is taken.
  - When ConfigurationDesk is started, it also uses the first free port from the specified range.
  - If ConfigurationDesk is not running or no MATLAB instance is running, then no ports from the specified range are used.
- 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 by the following prefixes: `HelpAbsLayer<xxx>`, `HelpInstaller<xxx>`.

## Using dSPACE Software on Virtual Machines (VMs)

### Introduction

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) prohibits:

- 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.

If you have any questions or encounter any problems, contact dSPACE Support ([www.dspace.com/go/supportrequest](http://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.

You can use Windows, Linux, or macOS® as the host operating system.

**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 |
|--------------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| AutomationDesk                       | —            | ✓ Known limitations: <ul style="list-style-type: none"> <li>▪ Access to DS1006 modular systems via dSPACE link boards is not possible.</li> <li>▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low.</li> <li>▪ Access to DS1104 R&amp;D Controller Boards is not possible.</li> </ul> | —          |
| Automotive Simulation Models         | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| Bus Manager (stand-alone)            | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| Calibration API package              | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| ConfigurationDesk                    | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| ControlDesk                          | —            | ✓ Known limitations: <ul style="list-style-type: none"> <li>▪ Access to DS1006 modular systems via dSPACE link boards is not possible.</li> <li>▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low.</li> <li>▪ Access to DS1104 R&amp;D Controller Boards is not possible.</li> </ul> | —          |
| Data Dictionary Manager              | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| DCI Configuration Tool               | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| dSPACE AUTOSAR Compare               | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| dSPACE Bus API Package               | —            | ✓ Known limitations: <ul style="list-style-type: none"> <li>▪ The usage of USB interfaces is not supported.</li> </ul>                                                                                                                                                                                                                                          | —          |
| dSPACE CAN API Package               | —            | ✓ Known limitations: <ul style="list-style-type: none"> <li>▪ The usage of USB interfaces is not supported.</li> </ul>                                                                                                                                                                                                                                          | —          |
| dSPACE ECU Flash Programming Tool    | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| dSPACE FlexRay Configuration Package | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| dSPACE Installation Manager          | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| ECU Interface Manager                | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| ECU bypassing target compiler        | ✓            | —                                                                                                                                                                                                                                                                                                                                                               | —          |
| Failure Simulation                   | —            | ✓ Known limitations: <ul style="list-style-type: none"> <li>▪ Access to DS1006 modular systems via dSPACE link boards is not possible.</li> </ul>                                                                                                                                                                                                               | —          |



| Product                              | Full Support | Support with Known Limitations                                                                                                                                                                                                                                                                                                                 | No Support      |
|--------------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
|                                      |              | <ul style="list-style-type: none"> <li>▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low.</li> <li>▪ Access to DS1104 R&amp;D Controller Boards is not possible.</li> </ul>                                                                                         |                 |
| Firmware Archives                    | —            | ✓ Known limitations:                                                                                                                                                                                                                                                                                                                           | —               |
| Firmware Manager                     |              | <ul style="list-style-type: none"> <li>▪ Access to DS1006 modular systems via dSPACE link boards is not possible.</li> <li>▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low.</li> <li>▪ Access to DS1104 R&amp;D Controller Boards is not possible.</li> </ul>     |                 |
| FPGA Programming Blockset            | —            | ✓ Known limitations:                                                                                                                                                                                                                                                                                                                           | —               |
|                                      |              | <ul style="list-style-type: none"> <li>▪ Access to DS1006 modular systems via dSPACE link boards is not possible.</li> <li>▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore the performance is very low.</li> <li>▪ Access to DS1104 R&amp;D Controller Boards is not possible.</li> </ul> |                 |
| Model Compare                        | ✓            | —                                                                                                                                                                                                                                                                                                                                              | —               |
| Model Container Utility              | ✓            | —                                                                                                                                                                                                                                                                                                                                              | —               |
| ModelDesk                            | —            | ✓ Known limitations:                                                                                                                                                                                                                                                                                                                           | —               |
|                                      |              | <ul style="list-style-type: none"> <li>▪ The Traffic Object Manager cannot show custom sensor points in the preview.</li> <li>▪ Plotting occasionally does not start if a start trigger is used.</li> </ul>                                                                                                                                    |                 |
| Model Interface Package for Simulink | ✓            | —                                                                                                                                                                                                                                                                                                                                              | —               |
| MotionDesk                           | —            | —                                                                                                                                                                                                                                                                                                                                              | ✓ <sup>1)</sup> |
| Platform API Package                 | —            | ✓ Known limitations:                                                                                                                                                                                                                                                                                                                           | —               |
|                                      |              | <ul style="list-style-type: none"> <li>▪ Access to DS1006 modular systems via dSPACE link boards is not possible.</li> <li>▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low.</li> <li>▪ Access to DS1104 R&amp;D Controller Boards is not possible.</li> </ul>     |                 |
| Real-Time Testing                    | —            | ✓ Known limitations:                                                                                                                                                                                                                                                                                                                           | —               |
|                                      |              | <ul style="list-style-type: none"> <li>▪ Access to DS1006 modular systems via dSPACE link boards is not possible.</li> <li>▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low.</li> <li>▪ Access to DS1104 R&amp;D Controller Boards is not possible.</li> </ul>     |                 |
| RTI Blocksets (Real-Time Interface)  | —            | ✓ Known limitations:                                                                                                                                                                                                                                                                                                                           | —               |
|                                      |              | <ul style="list-style-type: none"> <li>▪ Access to DS1006 modular systems via dSPACE link boards is not possible.</li> <li>▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore the performance is very low.</li> <li>▪ Access to DS1104 R&amp;D Controller Boards is not possible.</li> </ul> |                 |

| Product               | Full Support    | Support with Known Limitations | No Support |
|-----------------------|-----------------|--------------------------------|------------|
| Signal List Editor    | ✓               | —                              | —          |
| SYNECT                | ✓               | —                              | —          |
| SYNECT Server         | ✓               | —                              | —          |
| SYNECT License Server | ✓               | —                              | —          |
| SystemDesk            | ✓               | —                              | —          |
| TargetLink            | ✓               | —                              | —          |
| VEOS                  | ✓ <sup>2)</sup> | —                              | —          |

1) VMs do not fulfill the requirements for graphics adapters.

2) If you want to simulate adaptive AUTOSAR V-ECUs, refer to [Hypervisor Configuration for Adaptive V-ECU Simulation on Windows \(VEOS Manual !\[\]\(99f58673407353e96a019fbca558fd72\_img.jpg\)](#)).

### 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 using 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 !\[\]\(6059a5aa8b4ca7bb793408023d6c6e42\_img.jpg\)](#)). 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 the 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 licenses, 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 !\[\]\(9c2e8d1b5bd77cb5c9f83b7a9cff79fd\_img.jpg\)](#)).

**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 from the Windows Start menu (Start – Control Panel – Appearance and Personalization – Display – Adjust ClearType text).

---

### Using the 'Revert to snapshot' feature

#### NOTICE

##### Using the 'Revert to snapshot' feature causes licenses to become invalid.

If you use the 'Revert to snapshot' feature in a VM, all software-based CmContainers on your host PC (dSPACE Activation Container and/or dSPACE Borrow Container) become invalid and the contained licenses are lost.

- Do not use the 'Revert to snapshot' feature for VMs that contain software-based CmContainers with activated licenses.
- Store the license information on CmDongles. There, the CmContainers do not become invalid after use the 'Revert to snapshot' feature.

---

### Moving the virtual machine to a host PC with a different hardware configuration

#### NOTICE

##### Moving the virtual machine to a PC with a different hardware configuration causes licenses to become invalid.

If you move your virtual machine to a host PC with a different hardware configuration, all software-based CmContainers on your host PC (dSPACE Activation Container and/or dSPACE Borrow Container) become invalid and the contained licenses are lost. This happens, for example, if the CPU type of the physical PC changes.

- Do not move a virtual machine that contains software-based CmContainers with activated licenses to a host PC with a different hardware configuration.
- Store the license information on CmDongles. There, the CmContainers do not become invalid after the virtual machine is moved.
- However, if moving the virtual machine is absolutely necessary, contact dSPACE Support ([www.dspace.com/go/supportrequest](http://www.dspace.com/go/supportrequest)) beforehand to find solutions that can avoid major downtimes.

## 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 2023-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 hereafter. 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 <https://www.dspace.com/go/CompPatch>.
- RCP and HIL software products (of Release 2023-B) cannot be used in combination with RCP and HIL software products from earlier dSPACE Releases.

---

**Combining dSPACE products from earlier Releases**

For notes on the combined use of different products from earlier Releases, refer to [https://www.dspace.com/go/ds\\_sw\\_combi](https://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.

---

**Installing and running dSPACE software within the Windows service accounts**

Non-service-based dSPACE software is not designed to be installed or run in the context of any predefined Windows service account (LocalService, NetworkService, LocalSystem).

---

**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 that you close the dSPACE software manually before shutting down the PC.

---

**User Account Control**

It is recommended that you 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.

---

**Using high-DPI monitors**

Real-Time Testing has limitations for working with high-DPI monitors. The Real-Time Test Manager, the user interface for handling RTT sequences, does not support working with high-DPI monitors.

---

**High-contrast mode not supported**

dSPACE software was not developed for or tested against the Windows high-contrast mode. You are recommended to disable the high-contrast mode when working with dSPACE software. Otherwise, the visualization of the user interfaces might be impaired.

---

**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://docs.microsoft.com/en-us/windows/security/threat-protection/fips-140-validation>.

---

**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 <https://www.dspace.com/faq?346> or to the Microsoft Windows documentation.

---

**Settings in Windows for user locale and system locale must match**

MATLAB reads the user locale and system locale settings that are specified in Windows operating systems. The user locale and the system locale must match. If these settings are not the same, the system might not behave as expected when working with MATLAB and dSPACE software.

For instructions on checking and changing the settings, refer to [https://www.mathworks.com/help/matlab/matlab\\_env/setting-locale-on-windows-platforms.html?s\\_tid=gn\\_loc\\_drop](https://www.mathworks.com/help/matlab/matlab_env/setting-locale-on-windows-platforms.html?s_tid=gn_loc_drop).

This affects all MATLAB versions and all Windows operating systems, that are supported by dSPACE.

---

**Limitation for using single-user licenses**

If you use a PC with a Windows Server operating system, for example, Windows Server 2019, and you want to work with dSPACE software on this PC using Microsoft Remote Desktop Connection, you have to use a floating network license to activate license-protected software. Single-user licenses are not supported in this case.

## Limitations for Using Linux Features

---

**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:

- Ubuntu 20.04 LTS: <https://ubuntu.com/blog/fips-certification-ubuntu-20-04-lts>

---

**Long paths**

dSPACE software does not support the available path length of 4,096 characters. If a path that exceeds 260 characters is used directly or indirectly, the behavior of the dSPACE software is not defined.

---

**Locale**

dSPACE software was tested only on a system with a US English locale.

**A**

- ASM Battery Blockset
  - migration 42
  - new features 41
- ASM blocksets
  - migration 40
- ASM Diesel Engine Blockset
  - migration 43
- ASM Drivetrain Basic Blockset
  - migration 44
  - new features 44
- ASM Electric Components Blockset
  - migration 45
  - new features 45
- ASM Engine Gasoline Blockset
  - migration 48
- ASM Fuel Cell Blockset
  - migration 47
  - new features 46
- ASM Trailer Blockset
  - new features 50
- ASM Truck Blockset
  - new features 52
- ASM Vehicle Dynamics Blockset
  - migration 55
  - new features 54
- AutomationDesk
  - migration 38
  - new features 37

**B**

- Bus Manager (stand-alone)
  - migration 60
  - migration of projects and applications 60
  - new features 57
- Bus Manager in ConfigurationDesk
  - new features 71

**C**

- Common Program Data folder 12
- ConfigurationDesk
  - migration 77
  - new features 65
- ControlDesk
  - migration 102
  - new features 86

**D**

- DCI Configuration Tool
  - new features 107
- discontinuation 15
  - planned (hardware) 16
  - planned (software support) 15
  - software support 15
- Documents folder 12
- dSPACE ECU Flash Programming Tool
  - new features 109
- dSPACE FlexRay Configuration Package
  - new features 111
- dSPACE Python Extensions

- new features 115

- dSPACE XIL API
  - migration 118
  - new features 117

**E**

- ECU Interface Manager
  - migration 122
  - new features 121

**F**

- Firmware Manager
  - new features 125
- FPGA Programming Blockset
  - new features 127

**G**

- general enhancements and changes 13

**H**

- host PC software
  - Linux operating system 261
  - MATLAB 259
  - Windows operating system 261

**K**

- key features 20

**L**

- limitations for using Linux features 270
- limitations for using Windows features 268
- Linux
  - limitations 270
- Local Program Data folder 12

**M**

- MATLAB
  - requirements 259
  - supported releases 259
- MicroAutoBox III firmware
  - new features 133
- migration
  - ASM Battery Blockset 42
  - ASM blocksets 40
  - ASM Diesel Engine Blockset 43
  - ASM Drivetrain Basic Blockset 44
  - ASM Electric Components Blockset 45
  - ASM Engine Gasoline Blockset 48
  - ASM Fuel Cell Blockset 47
  - ASM Vehicle Dynamics Blockset 55
  - AutomationDesk 38
  - Bus Manager (stand-alone) 60
    - projects and applications 60
  - ConfigurationDesk 77
  - ConfigurationDesk projects and applications 80
  - ControlDesk 102
  - dSPACE XIL API 118
  - ECU Interface Manager 122

- Model and Sensor Interface 137
- Model Compare 136
- Model Interface Package for Simulink 142
- ModelDesk 140
- MotionDesk 143
- Real-Time Testing 145
- RTI 147
  - RTI Bypass Blockset 149
  - RTI CAN MultiMessage Blockset 151
  - RTI FPGA Programming Blockset 129
  - RTI LIN MultiMessage Blockset 153
  - SCALEXIO firmware 158
- migration of ConfigurationDesks projects and applications 80
- Model and Sensor Interface
  - migration 137
- Model Compare
  - migration 136
  - new features 135
- Model Interface Package for Simulink
  - migration 142
  - new features 141
- ModelDesk
  - migration 140
  - new features 139
- MotionDesk
  - migration 143
  - new features 143

**N**

- new features
  - ASM Battery Blockset 41
  - ASM Drivetrain Basic Blockset 44
  - ASM Electric Components Blockset 45
  - ASM Fuel Cell Blockset 46
  - ASM Trailer Blockset 50
  - ASM Truck Blockset 52
  - ASM Vehicle Dynamics Blockset 54
  - AutomationDesk 37
  - Bus Manager (stand-alone) 57
  - Bus Manager in ConfigurationDesk 71
  - ConfigurationDesk 65
  - ControlDesk 86
  - DCI Configuration Tool 107
  - dSPACE ECU Flash Programming Tool 109
  - dSPACE FlexRay Configuration Package 111
  - dSPACE Python Extensions 115
  - dSPACE XIL API 117
  - ECU Interface Manager 121
  - Firmware Manager 125
  - FPGA Programming Blockset 127
  - MicroAutoBox III firmware 133
  - Model Compare 135
  - Model Interface Package for Simulink 141
  - ModelDesk 139
  - MotionDesk 143
  - RTI CAN MultiMessage Blockset 151
  - RTI LIN MultiMessage Blockset 153
  - RTI Synchronized Time Base Manager Blockset 155
  - RTI/RTI-MP 147
  - RTLlib 147
  - SCALEXIO firmware 157

- Signal List Editor 161
- SystemDesk 168
- VEOS 251
- new hardware 14

**P**

- product overview 17

**R**

- Real-Time Testing
  - migration 145
- requirements
  - host PC Linux operating system 261
  - host PC software
    - MATLAB 259
  - host PC Windows operating system 261
- RTI Bypass Blockset
  - migration 149
- RTI CAN MultiMessage Blockset
  - migration 151
  - new features 151
- RTI FPGA Programming Blockset
  - migration 129
- RTI LIN MultiMessage Blockset
  - migration 153
  - new features 153
- RTI Synchronized Time Base Manager Blockset
  - new features 155
- RTI/RTI-MP
  - new features 147
- RTLib
  - new features 147

**S**

- SCALEXIO firmware
  - migration 158
  - new features 157
- Signal List Editor
  - new features 161
- supported MATLAB releases 259
- system requirements
  - Linux operating system 261
  - Windows operating system 261
- SystemDesk
  - new features 168

**T**

- TargetLink Data Dictionary
  - migration 200
    - discontinued documentation 200
    - manually upgrading libraries and models 203
    - upgrading existing data dictionaries 202

**V**

- VEOS
  - new features 251
- version history 17

**W**

- Windows
  - limitations 268