dSPACE Release

New Features and Migration

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

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

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

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

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

Software Updates and Patches

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

Important Notice

Contents

About This Document

Content

This document informs you about the new features of all the dSPACE software products in Release 2021-A. It also gives you an overview of software products with no or minor changes. There are instructions on migrating from earlier dSPACE Releases, especially from earlier product versions, if required.

Printed document

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

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

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

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

Special folders

Some software products use the following special folders:

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

```
%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>
```

or

```
%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>
```

Documents folder A standard folder for user-specific documents.

```
%USERPROFILE%\Documents\dSPACE\<ProductName>\
<VersionNumber>
```

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

```
%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\
<ProductName>
```

Accessing dSPACE Help and PDF Files

After you install and decrypt 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**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

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

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

Overview of dSPACE Release 2021-A

Introduction	Gives you an overview of the new key features in Release 2021-A and information about unchanged products.
Where to go from here	Information in this section

General Enhancements and Changes

Introduction	The following new features and changes concern several dSPACE products.
Python distribution	The support of Python 3.6 is discontinued with dSPACE Release 2021-A and is replaced by Python 3.9. For more information, refer to Migrating Python Scripts from Python 3.6 to Python 3.9 on page 27.
dSPACE certificate for dSPACE software	As of dSPACE Release 2020-B, all dSPACE executable files are digitally signed with a dSPACE certificate. To check this certificate, a valid chain of trust must be established on the PC on which the dSPACE software is installed. One part of the chain of trust is root certificates, which are available from trusted certification authorities and are installed by the Windows update mechanism.

Note

- If the dSPACE software is installed on a PC with Internet access, the root certificates are automatically installed on the PC by the Windows update mechanism. No further user actions are required.
- If your dSPACE software is to be installed on a PC without an Internet connection, you must install the root certificates manually on your PC. dSPACE strongly recommends installing the certificates before installing the dSPACE software. It is possible that the security software installed on your PC blocks the installation of the dSPACE software if no root certificates are available on your PC.

For instructions on manual installation, refer to Installing Root Certificates Required for dSPACE Software (Installing dSPACE Software 🕮).

Providing legal notes for using third-party software

Several dSPACE software products of dSPACE Release 2021-A 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.

Using dSPACE software on virtual machines (VM)

You can operate several dSPACE products on virtual machines. For more information, refer to Using dSPACE Software on Virtual Machines (VMs) on page 149.

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

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

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

Refer to Legacy Licensing Using CodeMeter Licenses (Working with CodeMeter Licensing Technology 🕮).

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

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

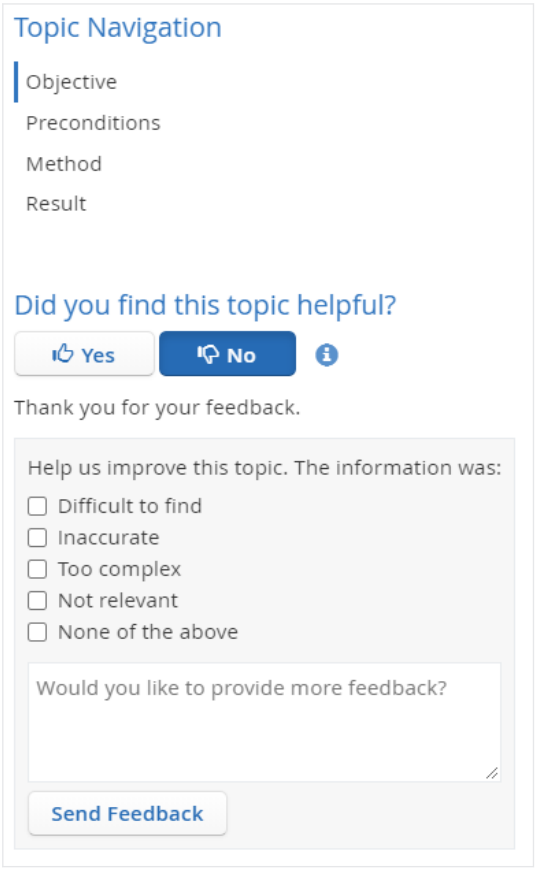
- MinGW (GNU Compiler Collection (GCC 6.3.0))
- Microsoft Visual Studio 2017 Professional

New Features of dSPACE Help

Feedback

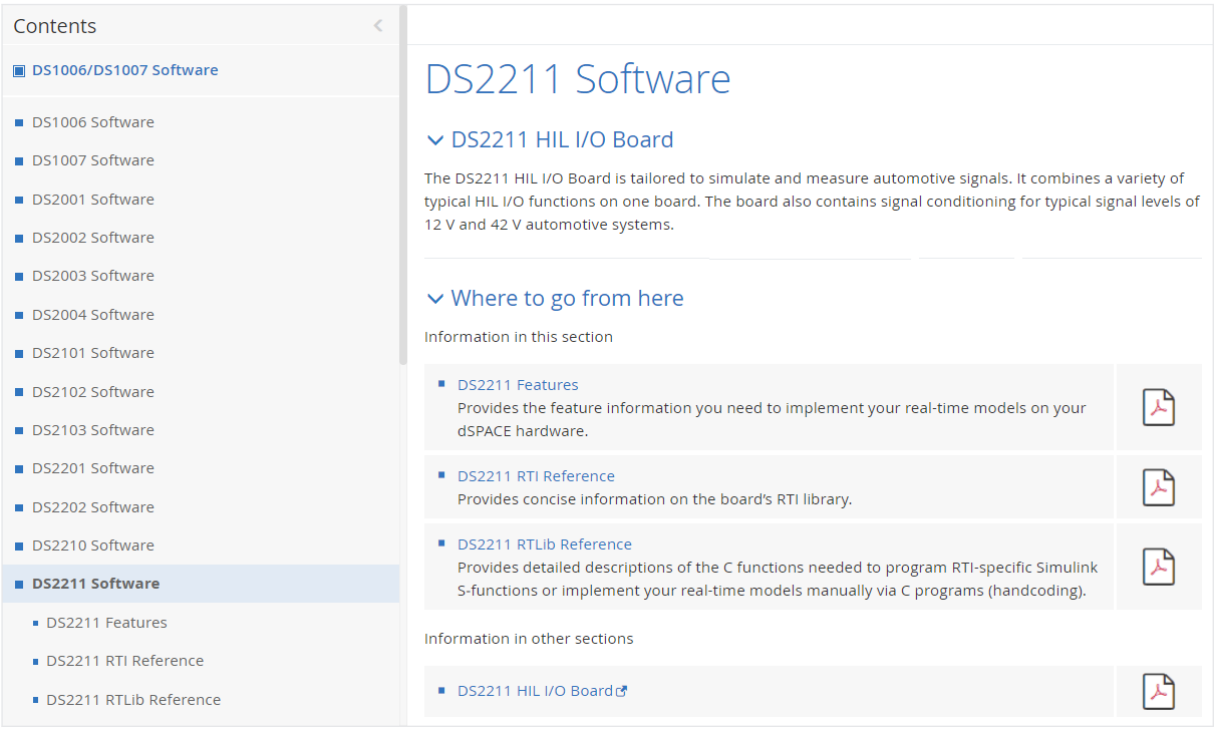
You can now send feedback on each dSPACE Help topic when your device is connected to the Internet.

Sending feedback does not collect, transmit, or use any personally identifiable information.



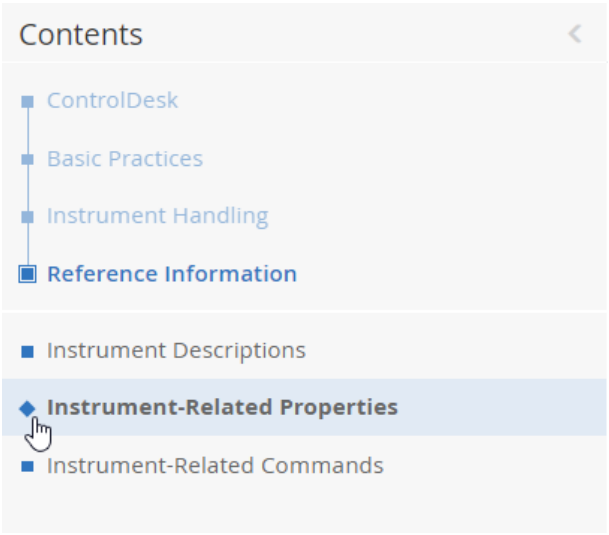
Enhanced content structure

The content structure of dSPACE Help was enhanced to improve navigation and user experience. For example, the new structure makes it easier for you to navigate to the desired I/O Board documentation.



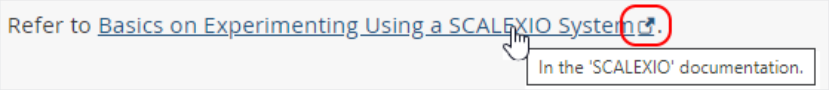
Enhanced content navigation

You can now close structure elements in the table of contents of the dSPACE Help, which improves content navigation.



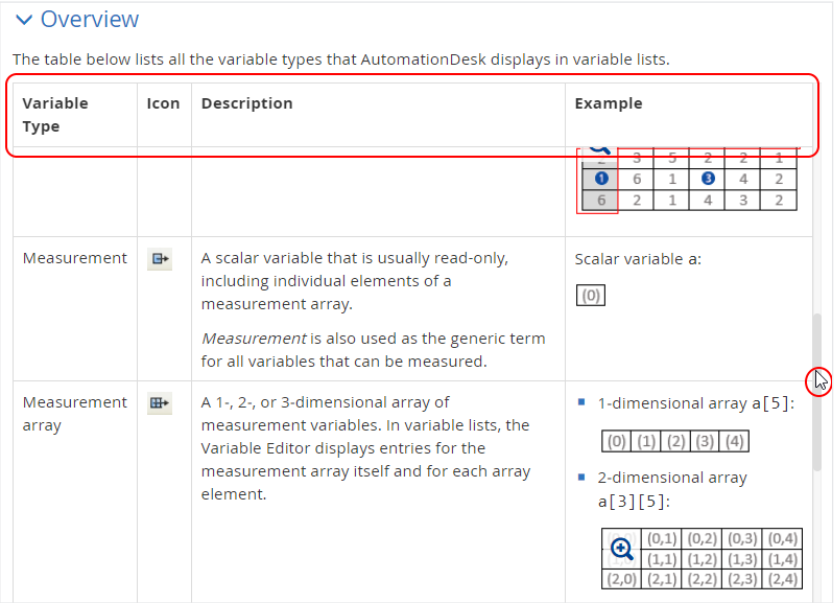
Visualization of external links

A link that points to a topic of another product is now visualized by a symbol. The related tooltip shows the target of the reference.



Fixed Table Header

In large tables, the table header remains visible when you scroll the table content.



Discontinuations

Introduction

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

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

Discontinuation of software support

Calibration Hub With dSPACE Release 2021-A, software support is discontinued for the Calibration Hub.

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

Discontinuation of .NET 2.0 APIs With dSPACE Release 2020-B, dSPACE discontinued the support for client programs and libraries built with .NET Runtime 2.0. This applies to any C#-based programs that use the provided

interfaces (APIs) for COM automation and for the Installation Manager. Applications using these interfaces have to use at least .NET Runtime 4.0.

Discontinuation of ConfigurationDesk options The following ConfigurationDesk options will be discontinued with an upcoming dSPACE Release:

- The **Generate Simulink Model Interface** command group in the context menu of a Simulink model in the **Model Browser**

  You can use the **Propagate to Simulink Model** and the **Generate New Simulink Model Interface** commands instead.
- The **Skip Model Code Generation** property of a Simulink model

  For Simulink models (SLX files) in ConfigurationDesk, Simulink® Coder™ checks whether more up-to-date code is available for the model and skips the code generation process if necessary. As an alternative, you can use Simulink implementation containers (SIC files) instead of SLX files. In this case, you can run a build process without generating code for a Simulink model. A MATLAB installation is therefore not required.

Planned discontinuation of dSPACE hardware

PHS-bus hardware As of December 2021, the hardware components for PHS-bus-based systems, such as the DS1006 Processor Board, the DS1007 PPC Processor Board, and all the PHS-bus I/O boards, will be discontinued. This also applies to the dSPACE Simulator Mid-Size and the dSPACE Simulator EcoLine. New Releases of dSPACE software will continue to support the PHS-bus hardware components until the end of 2023.

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

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			
	2019-B	2020-A	2020-B	2021-A
AutomationDesk	6.2	6.3	6.4	6.5 Refer to AutomationDesk on page 35.
Automotive Simulation Models	9.4	9.5	9.6	9.7 Refer to Automotive Simulation Models (ASM) on page 39.
Bus Manager (stand-alone)	6.4	6.5	6.6	6.7 Refer to Bus Manager (Stand-Alone) on page 53.

Product	dSPACE Release			
	2019-B	2020-A	2020-B	2021-A
ConfigurationDesk for RapidPro	6.4	6.5	6.6	6.7
ConfigurationDesk	6.4	6.5	6.6	6.7 Refer to ConfigurationDesk on page 57.
Container Manager	5.1	5.1	5.2	5.2
ControlDesk	7.1	7.2	7.3	7.4 Refer to ControlDesk on page 69.
DCI Configuration Tool	3.11	3.12	3.12	3.13 Refer to DCI Configuration Tool on page 77.
dSPACE AUTOSAR Compare	–	–	1.0	1.0
dSPACE CAN API Package	4.0.3	4.0.4	4.0.4	4.0.6
dSPACE ECU Flash Programming Tool	2.6	2.7	2.7	2.8
dSPACE FlexRay Configuration Package	4.4	4.5	4.6	4.7 Refer to dSPACE FlexRay Configuration Package on page 79.
dSPACE Installation Manager	5.4	5.5	5.6	5.7 Refer to dSPACE Installation Manager on page 81
dSPACE Python Extensions	3.2	3.3	3.4	4.0 Refer to dSPACE Python Extensions on page 83.
dSPACE XIL API .NET	2019-B	2020-A	2020-B	2021-A Refer to dSPACE XIL API .NET on page 85.
ECU Interface Manager	2.6	2.7	2.8	2.9 Refer to ECU Interface Manager on page 87.
Firmware Manager	3.0	3.1	3.2	3.3 Refer to Firmware Manager on page 91.
Model and Sensor Interface Blockset	–	–	1.0	1.1 Refer to Model and Sensor Interface Blockset on page 95.
Model Compare	3.0	3.0	3.1	3.1
ModelDesk	5.12	5.3	5.4	5.5 Refer to ModelDesk on page 99.
Model Interface Package for Simulink	4.2	4.3	4.4	4.5 Refer to Model Interface Package for Simulink on page 101.
MotionDesk	4.5	4.6	4.7	4.8 Refer to MotionDesk on page 103.
MotionDesk Blockset	2.6	2.6.1	2.6.2	2.6.3 Refer to MotionDesk on page 103.
Real-Time Testing	4.2	4.3	4.4	5.0 Refer to Real-Time Testing on page 107.

Product	dSPACE Release			
	2019-B	2020-A	2020-B	2021-A
RTI[1)]	7.13	7.14	7.15	7.16 Refer to RTI/RTI-MP and RTLib on page 109.
RTI-MP[2)]	7.13	7.14	7.15	7.16 Refer to RTI/RTI-MP and RTLib on page 109.
RTI Bypass Blockset	3.13	3.14	3.15	3.16 Refer to RTI Bypass Blockset on page 111.
RTI CAN Blockset	3.4.9	3.4.10	3.4.11	3.4.12
RTI CAN MultiMessage Blockset	5.3	5.4	5.5	5.6 Refer to RTI CAN MultiMessage Blockset on page 115.
RTI Electric Motor Control Blockset	1.4.1	1.4.1	1.4.2	1.4.3
RTI Ethernet Blockset	1.2.3	1.2.3	1.2.4	1.2.5
RTI Ethernet (UDP) Blockset	1.4.3	1.4.3	1.4.4	1.4.5
RTI FPGA Programming Blockset	3.8	3.9	3.10	3.11 Refer to RTI FPGA Programming Blockset on page 117.
RTI LIN MultiMessage Blockset	3.3	3.4	3.5	3.6 Refer to RTI LIN MultiMessage Blockset on page 121.
RTI RapidPro Control Unit Blockset	2.2.3	2.2.3	2.2.4	2.2.5
RTI Synchronized Time Base Manager Blockset	1.3	1.4	1.4.1	1.4.2
RTI USB Flight Recorder Blockset	1.2.2	1.2.2	1.2.3	1.2.4
RTI Watchdog Blockset	2.1.1	2.1.1	2.1.2	2.1.3
Sensor Simulation	1.2	1.3	1.4	1.5 Refer to Sensor Simulation on page 125.
SCALEXIO firmware	4.5	4.6	5.0	5.1 Refer to SCALEXIO Firmware on page 123.
SYNECT	2.8	2.9	2.10	2.11 Refer to SYNECT on page 127.
SystemDesk	5.4	5.4	5.5	5.5
TargetLink	5.0	5.0	5.1	5.1
Variable Editor[3)]	2.4	2.4	2.4	2.4
VEOS	4.5	5.0	5.1	5.2 Refer to VEOS on page 137.

1) Including the standard I/O blocksets.
2) Including the RTI Gigalink Blockset.
3) The Variable Editor is not part of the dSPACE Release DVD. It is available at https://www.dspace.com/go/requestreleasedownload.

If you have not performed regular updates, refer to the *New Features and Migration* documents for the dSPACE Releases listed above for information about the new features and 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 feature of AutomationDesk is:

- Support of Python 3.9.

For more information on the new features, refer to New Features of AutomationDesk 6.5 on page 35.

Bus Manager (stand-alone)

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

- New frame captures for capturing CAN frames that are received on the bus.
- Enhanced J1939 support
- Support of extended signal multiplexing that is specified in DBC communication matrices.
- New and enhanced bus configuration features
- Generation of optimized variable description files (TRC files)

For more information, refer to New Features of the Bus Manager (Stand-Alone) 6.7 on page 53.

ConfigurationDesk

The new key features of ConfigurationDesk are:

- The **Field-Oriented Control In/Out** function block is available to support E-Drive control I/O functionality on a SCALEXIO system.
- The **Lambda Probe In** function block is available to support engine control I/O functionality on the MicroAutoBox III.
- The following function blocks are available to support system functionality of the MicroAutobox III: **System Temperature Monitioring**, **USB Eject**, **Acceleration In**, **Atmospheric Pressure In**, **LED Out**.
- Various enhancements of the following function blocks: **FuSa System Monitoring**, **TCP**, **SPI Master**.
- Various enhancements of the Bus Manager for configuring bus communication for simulation, inspection, and manipulation purposes.

For more information, refer to ConfigurationDesk on page 57.

ControlDesk

The new key features of ControlDesk are:

Bus Navigator enhancements

- CAN: Display of the delta time (last frame)
- Context menu extensibility of Bus Navigator tree nodes

For more information, refer to New Bus Navigator Features (ControlDesk 7.4) on page 70.

ECU Diagnostics enhancements

- Support of CAN FD via built-in D-PDU API

For more information, refer to New ECU Diagnostics Features (ControlDesk 7.4) on page 71.

DCI Configuration Tool

The new key feature of the DCI Configuration Tool is:

- New command line option to generate ECU resets.

For more information on the new feature, refer to New Features of the DCI Configuration Tool 3.13 on page 77.

dSPACE FlexRay Configuration Package

The new key features of the dSPACE FlexRay Configuration Package are:

- Support of AUTOSAR Classic Platform R20-11.
- Support of static container IPDUs.
- Support of cryptographic IPDUs.

The new key features of the dSPACE FlexRay Configuration Tool are:

- Support of signals with opaque byte order format longer than 64 bytes.
- Support of BITFIELD-TEXTTABLE computation method.

For more information on the new features, refer to New Features of dSPACE FlexRay Configuration Package 4.7 on page 79.

dSPACE XIL API

The new key features of dSPACE XIL API and the Platform Management API are:

- Support of Python 3.9.
- dSPACE XIL API .NET on Linux is available for dSPACE Release 2021-A with enhanced platform support.

For more information on the new features, refer to New Features of dSPACE XIL API .NET 2021-A on page 85.

ECU Interface Manager

The new key features of the ECU Interface Manager are:

- Increased maximum number of ECU interface configurations that use the same interface type.
- Improved display of XCP service request messages.
- Enabling/disabling priority auto-decrementation.

For more information on the new feature, refer to New Features of ECU Interface Manager 2.9 on page 87.

Firmware Manager

The new key feature of the Firmware Manager is:

- Support of the KVM Hypervisor for the DS6001 Processor Board.

For more information on the new feature, refer to New Features of Firmware Manager 3.3 on page 91 and New Features of the SCALEXIO Firmware 5.1 on page 123.

MicroAutoBox III firmware

The new key features of the MicroAutoBox III firmware are:

- Extended DS1403 Processor Board support.
- Extended DS1554 Engine Control I/O Module support.

For more information on the new features, refer to New Features of the MicroAutoBox III Firmware 5.1 on page 93.

Model and Sensor Interface Blockset

You must use the Model and Sensor Interface Blockset for Sensor Simulation.

The blockset calculates and handles the simulation, status, feedback, and data for sensor failures between the simulation and the connected systems and applications. For example, you can connect, MotionDesk, the SensorSim application, and the Environment Sensor Interface Unit.

MotionDesk visualizes the simulation and the SensorSim application creates the sensor raw data.

The Environment Sensor Interface Unit separates the sensor raw data for each of the sensors from the SensorSim application and inserts it into the relevant sensor hardware.

You can also send data to simulate sensor failures to each of the sensors connected to the Environment Sensor Interface Unit. The blockset can also receive feedback information from the Environment Sensor Interface Unit and status information from the other connected systems or applications.

New features

- TX rate can be controlled through an inport signal in the Connection Settings block.
- Multiple inlined objects for a single transformation object can be specified in the Simulation Data Objects block.
- Sensor sphere support using a position simulation data object type.
- Improvements to the automation API to support multiple objects.
- Block priorities are implemented that define the execution order.
- Improved handling of connected signals and error detection.
- Improved user interface control and error logging.
- Improved demos.

For more information on the new features, refer to New Features of Model and Sensor Interface Blockset 1.1 on page 95.

Model Interface Package for Simulink

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

- Support of MATLAB R2021a.
- Support of TRC file entries for model workspace parameters of referenced models.

For more information on the new features, refer to Model Interface Package for Simulink on page 101.

ModelDesk

The new key features of ModelDesk are:

- Support of Python version 3.9.
- Parameterization: You can add buttons on parameter pages to start user-defined Python scripts.

For more information on the new features, refer to New Features of ModelDesk 5.5 on page 99.

MotionDesk

The new key features of MotionDesk are:

- **Installation**: MotionDesk is installed in its own folder structure in the selected installation path.
- **Bird's-eye view**: The bird's-eye view can be enabled for the selected view. This is a two-dimensional view above the scene.
- **Scene editing**: You can edit the scene without the dedicated scene edit mode. For example, you can freely move, resize, or rotate an object in the scene in the **3-D View**. You can also copy single and groups of objects in the scene and paste them to another location.
- **3-D object library vehicles**: New vehicles and trailers have been added to the 3-D library.
- **Animated characters**: The speed of the movements of the animated characters changes with the changes in the simulation.
- **Scene generation**: The generation of the road network geometries supports OpenDRIVE roads.

For more information on the new features, refer to New Features of MotionDesk 4.8 on page 103.

Python Extensions

The new key feature of Python Extensions is:

- Support of Python 3.9.

For more information, refer to dSPACE Python Extensions on page 83.

Real-Time Testing

The new key features of Real-Time Testing are:

- Support of Python 3.9 on the host PC.
- Real-Time Testing can be run on a Linux operating system.

For more information on the new features, refer to New Features of Real-Time Testing 5.0 on page 107.

RTI, RTI-MP, and RTLib

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

- Support of MATLAB R2021a.

For more information, refer to New Features of RTI/RTI-MP and RTLib on page 109.

RTI Bypass Blockset

The new key features of the RTI Bypass Blockset are:

- Configuration of compilers in on-target bypassing models.
- Support of Green Hills compilers for on-target bypassing.

For more information on the new features, refer to New Features of the RTI Bypass Blockset 3.16 on page 111.

RTI CAN MultiMessage Blockset

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

- Support of AUTOSAR Classic Platform R20-11.
- Support of further functions of the Bus Custom Code interface.
- New lesson in RTI CAN MultiMessage Blockset Tutorial on E2E protection according to AUTOSAR E2E profiles.

For more information on the new features, refer to New Features of the RTI CAN MultiMessage Blockset 5.6 on page 115.

RTI FPGA Programming Blockset

The new key features of the RTI FPGA Programming Blockset 3.11 are:

- Extended Xilinx® software support.
- Enhancements to the FPGA framework for the MicroAutoBox III and SCALEXIO.
- Enhancements to the script interface.

For more information on the new features, refer to New Features of the RTI FPGA Programming Blockset 3.11 on page 117.

RTI LIN MultiMessage Blockset

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

- Support of AUTOSAR Classic Platform R20-11.

For more information on the new feature, refer to New Features of the RTI LIN MultiMessage Blockset 3.6 on page 121.

Sensor Simulation

Sensor Simulation lets you validate camera, fish-eye, laser, lidar, and radar sensors.

A number of performance optimizations have been added to the SensorSim application for road shapes, lane markings, and fish-eye sensors for sensor simulation.

For more information on the new features, refer to New Features of Sensor Simulation 1.5 on page 125.

SYNECT

The new key features of SYNECT 2.11 are:

- SYNECT IIS assistant to simplify server application hosting.
- Dashboards to present test management data in browsers.

For more information on the new features, refer to New Features of SYNECT 2.11 on page 128.

VEOS

The new key features of VEOS are:

- Modeling extensions and enhancements
- Running VEOS in a Linux Docker container
- Adaptive V-ECU support enhancements
- Command line interface extensions

For more information on the new features, refer to New Features of VEOS 5.2 on page 137.

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 2021-A

Introduction

After you install Release 2021-A, some additional steps might be required.

Migrating from dSPACE Release 2020-B

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

Message Reader API The Message Reader API is available for several dSPACE products. For information on the related migration issue, see below.

Message Reader API change in dSPACE Release 2021-A

There is a migration issue specific to the Message Reader API. The issue occurs if you use the API with Python. The issue was caused by the migration to Python 3.9/pythonnet 2.5.3 with dSPACE Release 2021-A.

There is no migration issue to consider if you use the API with C#.

Specifying a product filter As of dSPACE Release 2021-A, the `Products` property of the `MessageReaderSettings` class can no longer be used to set the list of products for which to filter in the log sessions. The Message Reader API provides the `SetProducts` method for this purpose.

The following table shows how to specify a product filter before and after migration:

Using Message Reader API of ...	
... dSPACE Release 2020-B and Earlier (Python 3.6)	**... dSPACE Release 2021-A and Later (Python 3.9)**

`# Specify products whose messages to read:`
`Settings = MessageReaderSettings()`
`Settings.Products.Add('ControlDesk')`
`Settings.Products.Add('AutomationDesk')`
	`# Specify products whose messages to read:`
	`Settings = MessageReaderSettings()`
	`Settings.SetProducts(['ControlDesk', 'AutomationDesk'])`

Migrating from dSPACE Release 2020-A or earlier

To migrate from dSPACE Release 2020-A or earlier to Release 2021-A, 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 2021-A.

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 2021-A\Print\PreviousReleases`.
- On the dSPACE DVDs. Refer to `\Doc\PreviousReleases`.
- At www.dspace.com/go/migration for download. Here, you can also find *New Features and Migration* documents for very early Releases.

Migrating Python Scripts from Python 3.6 to Python 3.9

Where to go from here

Information in this section

Main Changes in Python 3.9

Main reason for using Python 3.9 instead of Python 3.6

The end of life of Python 3.6 is scheduled for the end of December 2021. Therefore, dSPACE switches to Python 3.9 with dSPACE Release 2021-A.

Related documents available on the Python website

For a short overview of the main changes in Python 3.9, refer to Technical Changes on page 30. For information on all the changes in the Python language and environment from Python 3.6 to Python 3.9, refer to Python Software Foundation at www.python.org.

The Python Software Foundation provides the following documents:

- What's New from Python 3.6 to 3.7:
  https://docs.python.org/3.7/whatsnew/3.7.html
- What's New from Python 3.7 to 3.8:
  https://docs.python.org/3.8/whatsnew/3.8.html
- What's New from Python 3.8 to 3.9:
  https://docs.python.org/3.9/whatsnew/3.9.html

Related topics

Basics

Technical Changes 30

Main Changes in Handling Python 3.9 with dSPACE Software

Introduction

With Python 3.9, some aspects of handling Python with dSPACE software have changed.

Libraries

The libraries and components used with Python 3.9 and distributed on dSPACE DVDs have changed as shown in the following table.

Package	Python 3.6 (Release 2020-B)	Python 3.9 (Release 2021-A)
comtypes	1.1.7	1.1.7
Core	3.6.8	3.9.1
cycler	0.10.0	0.10.0
future	0.18.2	0.18.2
grpcio	1.29.0	1.34.0
grpcio_tools	1.29.0	1.34.0
kiwisolver	1.2.0	1.3.1
lxml	–	4.6.2
matplotlib	3.2.1	3.3.3
numpy	1.18.4	1.19.3
pillow	7.1.2	8.0.1
pip	20.1.1	20.3.1

Package	Python 3.6 (Release 2020-B)	Python 3.9 (Release 2021-A)
protobuf	3.12.2	3.14.0
pycparser	–	2.20
pyglet	1.5.5	1.5.11
pyparsing	2.4.7	2.4.7
pypubsub	4.0.3	4.0.3
Python-dateutil	2.8.1	2.8.1
pythonnet	2.4.2	2.5.3
pytz	2020.1	2020.4
pywin32	227.10	300.10
scipy	1.4.1	1.5.4
six	1.15.0	1.15.0
wxPython	4.1.0	4.1.1
yapsy	1.12.2	1.12.2

General Information on Using Python Installations

Introduction

If you work with dSPACE software from dSPACE Release 2020-B or earlier, which supports Python 3.6, and dSPACE software from dSPACE Release 2021-A or later, which supports Python 3.9, both Python versions are installed on the PC and can be used in parallel.

Limitations when using Python 3.6 and Python 3.9 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 2021-A including SystemDesk, Python 3.6 will be installed last. Therefore, PythonWin 3.6 is registered to open Python files. To set the file associations to PythonWin 3.9, execute the `PythonInstaller.exe` from `<DVDRoot>\Products\Common\Python3.9` again or use the **`Edit with PythonWin 3.9`** command on the context menu of a Python 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/3.9/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 3.6 are available up to and including Release 2020-B.

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 3.9, might cause a conflict.

Technical Changes

Introduction

Migrating from Python 3.6 to Python 3.9 has caused minor 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 Python 3.9 require manual migration of your Python scripts.

Python.NET: Return value of IList type is converted to PyList Since Python.NET 2.4, return values of `SystemCollections.Generic.IList` type have been converted to the `PyList` type. Because some methods and properties of an `IList` object are not available with a `PyList` object, dSPACE provided the modified Python.NET 2.4.1 and 2.4.2 packages, which suppressed the conversion.

As of Python.NET 2.5.3, the modifications by dSPACE are no longer provided. You must therefore migrate the methods and properties of an `IList` object to

the related methods and properties of a `PyList` object. For example, you must use `len(MyObject)` instead of `MyObject.Count`.

Note

The created Python list is a copy of the .NET object. Modifications to the Python list therefore have no effect on the associated .NET object. To apply the modifications to the .NET object, assign the Python list afterward.

ctypes: Loading library after changing the PATH variable Loading a library via `ctypes` can fail if the path to the library was added to the PATH variable beforehand.

For loading a library in a safe manner, use `ctypes.WinDLL(NameOfDll, winmode=8)` instead of `ctypes.windll.LoadLibrary(NameOfDll)`.

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 - MAPort, 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 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 3.9:
  - 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
- Python 2.7.11:
  - All other real-time platforms

You must migrate your custom scripts to Python 3.9 manually. Real-Time Testing does not provide any features to automatically migrate custom scripts from Python 3.6 to Python 3.9.

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

Using dSPACE XIL API .NET

All dSPACE XIL API .NET methods and properties returning `IList` types must be migrated to the `PyList` type except for the `ASAM.XIL.Interfaces.Testbench.Common.Capturing.ICapture.Variables` property, which uses the `ObservableList` type.

Using SYNECT

Changed list handling of Python plug-ins The handling of lists in Python plug-ins changed. However, you can import the `ListAsRawEncoder` module to migrate scripts.

This is due to changes of list handling that were introduced with Python.NET 2.5.3. For details, refer to Technical Changes on page 30.

Setting Min/Max properties of variables The handling of variables when exchanging data via Python plug-ins changed. You now have to convert strings to double values if you specify the Min/Max properties of variables.

Old listing:

```
Variable = SignalsAndParameterManagement.Variable()
variable.Min = '-10'
variable.Max = '10'
```

New listing:

```
Variable = SignalsAndParameterManagement.Variable()
variable.Min = float('-10')
variable.Max = float('10')
```

For details on the variable class for exchanging signals and parameters, refer to Variable (SYNECT Guide 🕮).

Using SystemDesk

SystemDesk uses an external Python interpreter. It is therefore not affected by the Python migration. Only the demos depends on the installed Python distribution. Because SystemDesk is not updated with dSPACE Release 2021-A,

migrated demos will be available with a patch installation, refer to http://www.dspace.com/go/patches.

AutomationDesk

Where to go from here

Information in this section

New Features of AutomationDesk 6.5

General enhancements

Python 3.9 support Python scripts that are used in AutomationDesk projects are automatically migrated when you open the projects. For the required manual migration steps and general information on the migration, refer to Migrating Python Scripts from Python 3.6 to Python 3.9 on page 27.

Migrating to AutomationDesk 6.5

General migration aspects

If you open an AutomationDesk project with a later AutomationDesk version, the software automatically detects whether migration is required. For AutomationDesk projects that are saved in the legacy format, i.e., that were created with AutomationDesk 6.1 or earlier, you must click **OK** in the message dialog to start the migration. The migration of AutomationDesk projects in the new XML format does not have to be confirmed. Save the migrated project to another path or name.

An AutomationDesk project loaded and saved with AutomationDesk 6.3 and later can be opened in AutomationDesk 6.2. The different schema versions are

then displayed in a warning message. It might not be possible to execute the project with the earlier AutomationDesk version.

Note

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

- You must create backups of the project and of the linked custom libraries.
- AutomationDesk must be running properly. The Log Viewer must not display any error messages.
- The built-in libraries, required custom libraries, and other packages must be loaded properly.

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

For more information, refer to Migrating AutomationDesk (AutomationDesk Introduction And Overview 🕮).

Migration to the new serialization format introduced with AutomationDesk 6.2

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

If you use a version control system, you can get and open projects and custom libraries in the legacy formats. They are automatically migrated to the new serialization format and lose their connections to the version control system. You must add the migrated projects and custom libraries to new version control projects to avoid mixing the formats in the repository. For more information, refer to How to Migrate Projects or Custom Libraries Under Version Control (AutomationDesk Basic Practices 🕮).

Project import restricted as of Release 2021-A

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

Discontinued libraries and blocks

If you open a project containing discontinued elements in AutomationDesk 6.4, the discontinued data objects are replaced by **Discontinued data object** data objects and the discontinued blocks are replaced by **Discontinued block** blocks during the automatic project update. This lets you load your projects and search for blocks and data objects to be migrated. If you execute a project containing elements of a discontinued library via AutomationDesk or an API script, project execution will stop with an exception.

The following blocks and data objects are affected.

MATLAB Access Affected blocks are:

- HideApplication
- ShowApplication
- MinimizeCommandWindow
- RestoreCommandWindow

For more information on migration aspects, refer to Migrating AutomationDesk (AutomationDesk Introduction And Overview 🕮).

Planned discontinuations

RS232 library The RS232 library is included in AutomationDesk 6.5 for the last time. For later versions of AutomationDesk, a custom library that provides the same functionality will be available on demand. Refer to https://www.dspace.com/go/AUD-RS232-CustLib.

Python scripts using the `rs232lib2` Python module in an **Exec** or **ExecFile** block will also be affected by the discontinuation. As an alternative, you can use the standard Python serial port extension `pyserial`.

Automation Server As of December 2022, the *Automation Server* license will be discontinued. New Releases of the AutomationDesk Automation Server will be available for customers with a Software Maintenance Service Contract until June 2022.

The *AutomationDesk Basic* license provides the technical equivalent to the Automation Server.

For more information, refer to http://www.dspace.com/go/discontinuation.

Automotive Simulation Models (ASM)

Where to go from here

Information in this section

All ASM Products

New Features of All ASM Blocksets

Tutorial videos

There are now ASM tutorial videos on the dSPACE website. The videos describe the first steps with ASM.

To access the videos, go to https://www.dspace.com/en/pub/home/medien/videos.cfm.

ASM Base InCylinder

New Features of ASM Base InCylinder Blockset 2.7.1

EGR_COOLER block	An internal reset to the **Unit Delay** block has been added.
INTERCOOLER block	An internal reset to the **Unit Delay** block has been added.

ASM Drivetrain Basic

Migrating to ASM Drivetrain Basic Blockset 5.4.1

FUEL_CONSUMPTION block	A line break has been removed from the internal **Coasting** subsystem.

Related topics

Basics

Migrating ASM Models (ASM User Guide 🕮)

ASM Electric Components

Where to go from here

Information in this section

New Features of ASM Electric Components Blockset 3.11.1

BLDC_CONTROLLER_BASIC block	You can now configure the controller for torque control. The output of the speed controller is a set point for the torque and no longer a set point for the current.
BLDC_CONTROLLER block	You can now configure the controller for torque control. The output of the speed controller is a set point for the torque and no longer a set point for the current.
SCIM_CONTROLLER_BASIC block	You can now configure the controller for torque control. The output of the speed controller is a set point for the torque and no longer a set point for the current.
SCIM_CONTROLLER block	You can now configure the controller for torque control. The output of the speed controller is a set point for the torque and no longer a set point for the current.
PMSM_D_Q_NONLINEAR block	The JMAG converter for generating a ModelDesk parameter file and a MATLAB M-file with all machine parameters of the **ASM PMSM nonlinear** model from a JMAG-RT file was removed for compatibility reasons. For converting JMAG parameters, contact the ASM support.

Migrating to ASM Electric Components Blockset 3.11.1

BLDC_CONTROLLER_BASIC block	Four new inports have been added to the block: ▪ Const_Back_EMF[Vs] ▪ Const_Limit_Trq[Nm]

- Const_Polepairs[-]
- Trq_Set[Nm]

BLDC_CONTROLLER block

A new inport has been added to the block: Trq_Set[Nm]

Three new parameters have been added to the block:

- Const_Back_EMF[Vs]
- Const_Limit_Trq[Nm]
- Const_Polepairs[-]

SCIM_CONTROLLER_BASIC block

Two new inports have been added to the block:

- Const_Limit_Trq[Nm]
- Trq_Set[Nm]

SCIM_CONTROLLER block

A new inport has been added to the block: Trq_Set[Nm]

A new parameter has been added to the block: Const_Limit_Trq[Nm]

Related topics

Basics

Migrating ASM Models (ASM User Guide 🕮)

ASM Environment

New Features of ASM Environment Blockset 4.14

MANEUVER_FCN_CALL_ GENERATOR block

This block is new. It creates function call events separated by a sample-based gap.

If specific maneuver state transitions occur, the block internally creates a reset pulse to synchronize the generation of function call events with the start of the maneuver. These function calls are used, e.g., in an ASM Traffic demo model to synchronize the execution of the **MotionDesk Interface** subsystem with the start of the maneuver.

ASM Gasoline Engine Basic

Discontinuation of the ASM Gasoline Engine Basic Blockset

ASM Gasoline Engine Basic blockset	The ASM Gasoline Engine Basic blockset is discontinued. Use the ASM Gasoline Engine blockset instead. It provides all the functionalities of the ASM Gasoline Engine Basic blockset. For more information, refer to https://www.dspace.com/go/EndOfLifeASMGasolineEngineBasic.
Model migration	The migration of ASM Gasoline Engine Basic models is not supported.
ModelDesk support	ModelDesk will not migrate ASM Gasoline Engine Basic files. However, you can still use the remaining project.

ASM Pneumatics

Where to go from here

Information in this section

New Features of ASM Pneumatics Blockset 2.0.13

Resettable Memory blocks	All Memory and Unit Delay blocks of the library are now resettable because the MemoryWithReset block of the ASM Utils library is now used.
DetectFirstSampleHit block	The block has been added and linked to several blocks of the library to avoid multiple instances of identical blocks.
ExternalSignalInitialization block	The block has been added and linked to several blocks of the library to avoid multiple instances of identical blocks.
Volume_AirSpring block	The block has been added and linked to several blocks of the library to avoid multiple instances of identical blocks.
Volume_LiftBellow block	The block has been added and linked to several blocks of the library to avoid multiple instances of identical blocks.
Inversion of sign convention	The sign convention of all Displ_AirBellow and Displ_LiftBellow signals in the library has been inverted. Now, positive displacements mean compressed air/lift bellows.

Changes in the ASM Pneumatic Demo Models

Added reset signal

The Reset_VehicleStates[0|1] signal is now routed to library blocks in the demo model to enable a simultaneous reset of all states.

Extension of Disp_Plant

In the TruckPneumatics modules, scopes have been added to the UserInterface/Disp_Plant/Suspension subsystem to display state variables of the second rear axle of the truck.

Migrating to ASM Pneumatics Blockset 2.0.13

Reset inport

There is a new inport: Reset[0|1]

This applies to the following blocks:

- SOFT_ECU_EPM_CONTROL_FRONT
- SOFT_ECU_EPM_CONTROL_REAR
- SOFT_ECU_EPM_CONTROL_REAR_2ND
- SOFT_ECU_EPM_CONTROL_REAR_3RD
- SOFT_ECU_EPM_CONTROL_REAR_4TH
- TRL_CONTROL_MODULATOR_CTRL
- TRL_SOFT_ECU_EPM_CONTROL_FRONT
- TRL_SOFT_ECU_EPM_CONTROL_REAR
- TRL_SOFT_ECU_EPM_CONTROL_REAR_2ND

Displ_Spring inport

A Gain block with factor *minus one* is applied to the Displ_Spring inport.

This applies to the following blocks:

- AIR_SUSPENSION_FORCES_FRONT
- AIR_SUSPENSION_FORCES_REAR
- AIR_SUSPENSION_FORCES_REAR_2ND
- AIR_SUSPENSION_FORCES_REAR_3RD
- AIR_SUSPENSION_FORCES_REAR_4TH
- TRL_AIR_SUSPENSION_FORCES_FRONT

- TRL_AIR_SUSPENSION_FORCES_REAR
- TRL_AIR_SUSPENSION_FORCES_REAR_2ND

Related topics

Basics

Migrating ASM Models (ASM User Guide 🕮)

ASM Utils

New Features of ASM Utils 4.2.1

ASM_FCN_CALL_GENERATOR block	This block has been added to the ASM Utils library. It creates function call events separated by a defined sample-based gap. If the Reset input is set to *1*, the block immediately generates a function call event and then continues generating function call events with the defined sample-based gap.
Maneuver control	The implementation of the maneuver control has been extended. Now, a *start, stop, or reset command* that is triggered by the user via the ModelDesk Maneuver Control button, will be considered even if a VEOS application is paused or blocked, e.g., during co-simulation. The command will be executed as soon as the application switches to the *run* state again. This applies to the following commands: ▪ MANEUVER_START ▪ MANEUVER_STOP ▪ RESET

ASM Vehicle Dynamics

Migrating to ASM Vehicle Dynamics Blockset 4.2.1

FUEL_CONSUMPTION block	A line break has been removed from the internal **Coasting** subsystem.

Related topics

Basics

Migrating ASM Models (ASM User Guide 🕮)

Bus Manager (Stand-Alone)

Where to go from here

Information in this section

New Features of the Bus Manager (Stand-Alone) 6.7

Capturing CAN frames

The Bus Manager (stand-alone) now lets you capture frames that are received on a CAN bus. Capturing CAN frames is independent from communication matrices. Instead, you can capture CAN frames via frame capture elements, which you can add to the **Inspection** part of a bus configuration. The frame capture elements let you capture any CAN frame that is received in a communication cluster. Additionally, you can specify filters to capture specific frames. Via the **Frame Capture Data** feature, you can access the captured frame data, e.g., to directly provide diagnostic frames and network management frames to a mapped behavior model.

For more information, refer to Capturing CAN Frames (Bus Manager (Stand-Alone) Implementation Guide 🕮).

Enhanced J1939 support

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

Support of dynamic network management according to J1939-81 Dynamic network management according to J1939-81 is supported. Via the **J1939 Network Management Enable** bus configuration feature, you can enable dynamic network management for a J1939 network node. If you do this, the network node can send and respond to received address claims. Depending on the setting of its `arbitrary address capable`

parameter and its priority, the network node can claim a new address in case of an address conflict.

For more information, refer to Enabling and Disabling J1939 Network Management (Bus Manager (Stand-Alone) Implementation Guide 🕮).

Support of missing sending and/or receiving ECUs such as bridge ECUs J1939 communication that is specified in a DBC communication matrix is supported even if the sending and/or receiving ECU is not available in the DBC file. For example, this can happen if bridge ECUs are used to exchange J1939 communication between different communication clusters. In this case, the bridge ECU routes J1939-compliant PDUs from one communication cluster to the other by using the transport protocol address of the original sending and/or receiving ECU, which is not available in the communication matrix. In this case, the Bus Manager generates the required ECUs, names them `DS_UnknownJ1939ECU_<transport protocol address in hex>`, and assigns the affected PDUs to the related ECUs.

For more information, refer to Working with Communication Matrices (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Manipulation of J1939-compliant PDUs with up to 8 bytes J1939-compliant PDUs with up to 8 bytes are supported for manipulation. You can assign such PDUs to the **Manipulation** part of a bus configuration and manipulate the PDUs and their ISignals by using bus manipulation features.

For an overview of available bus manipulation features, refer to Basics on Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮). For basic information on using bus manipulation features, refer to Basics on Bus Manipulation Features (Bus Manager (Stand-Alone) Implementation Guide 🕮).

Configurable payload length of J1939-compliant PDUs You can configure the payload length of J1939-compliant PDUs that is specified in the communcation matrix. Refer to Configurable Settings of PDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮).

Support of extended signal multiplexing

The Bus Manager (stand-alone) now supports extended signal multiplexing for CAN communication that is specified in DBC communication matrices. In general, the value of a *multiplexer signal* can determine that a *multiplexed signal* is included in an `extended multiplexed IPDU`. With extended signal multiplexing, a multiplexed signal itself can serve as a multiplexer signal, i.e., its value can determine other multiplexed signals to be included in the extended multiplexed IPDU.

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

New bus configuration features

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

Bus Configuration Feature	Bus Configuration Part	Purpose	Further Information
PDU Length	Simulated ECUs	Lets you access the payload length of CAN PDUs at run time, e.g., to simulate dynamic length CAN PDUs.	Accessing the Payload Length of CAN PDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮)
SecOC Freshness Overwrite Value	Manipulation	Lets you manipulate the freshness value of secured IPDUs with a user-defined freshness value.	Overwriting the Freshness Value of Secured IPDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Generating optimized variable description files (TRC files)

The Bus Manager (stand-alone) now lets you generate an optimized set of variables into the variable description file (TRC file). In general, variables are generated into the variable description file for all function ports with enabled test automation support. By using the new **Variable description** property of each bus configuration, you can generate an optimized set of variables for function ports of the bus configuration whose test automation support is enabled. This can significantly improve run-time performance. For newly added bus configurations, the optimized set of variables is generated by default.

For more information, refer to Accessing Function Ports with Enabled Test Automation Support in Variable Description Files (Bus Manager (Stand-Alone) Implementation Guide 🕮).

General enhancements

The Bus Manager (stand-alone) now provides the following general enhancements:

- The performance for working with communication matrices and bus configurations in the ConfigurationDesk application and for executable applications is improved. This is achieved by various enhancements. For example:
  - By discarding unsupported communication matrix elements during the import of a communication matrix. Refer to Working with Communication Matrices (Bus Manager (Stand-Alone) Implementation Guide 🕮)
  - By disabling the evaluation of Bus Manager-related conflicts by default. Refer to Basics on Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮).
  - By improving the internal data model of the Bus Manager.
- CAN frames with a payload length of 0 bytes are supported.

- The `RX-ACCEPT-CONTAINED-IPDU` AUTOSAR attribute of container IPDUs with a dynamic container layout is evaluated. The setting of this attribute determines which IPDUs are accepted as contained IPDUs. Refer to Aspects of Supported AUTOSAR Features (Bus Manager (Stand-Alone) Implementation Guide 🕮).

  Tip

  If the `RX-ACCEPT-CONTAINED-IPDU` attribute of a container IPDU is set to `ACCEPT-CONFIGURED`, the header ID of the related contained IPDUs has to be unique only for the container IPDU and not for the entire communication matrix.

- Signals whose signal length is a multiple of 8 and exceeds 64 bits are imported as UInt8[] array signals with static length.
- Initial values that are specified in communication matrices, e.g., initial ISignal values, not only apply to the **Initial value** property of applicable bus configuration function ports, but also to the **Initial substitute value** property.

New supported AUTOSAR Release

The Bus Manager (stand-alone) now supports AUTOSAR system description files based on AUTOSAR Classic Platform Release R20-11.

Support of further functions of the Bus Custom Code interface

The Bus Manager (stand-alone) now supports the following functions of the Bus Custom Code interface: `DsBusCustomCodePdu_GetCanChannelName`, `DsBusCustomCodePdu_GetCanFrameTriggering`, `DsBusCustomCodePdu_GetLinFrameTriggering`, `DsBusCustomCodeSecOCPduFeature_setAuthenticatorPositionOffset`, and `DsBusCustomCodeSecOCPduFeature_getAuthenticatorPositionOffset`

For more information, refer to Overview of the Handles and Their Functions (Bus Custom Code Interface Handling 🕮).

Migrating to Bus Manager (Stand-Alone) 6.7

Working views exported with Bus Manager (stand-alone) 6.6 or earlier

The internal data model of bus configurations has changed. Therefore, working views (CAFX files) that contain **Bus Configuration** function blocks and that were exported with Bus Manager (stand-alone) 6.6 or earlier are no longer supported. If you try to import such a CAFX file, the import is aborted and a warning message is displayed.

ConfigurationDesk

Introduction

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

Where to go from here

Information in this section

New Features of ConfigurationDesk 6.7

Extended support of SIC file versions

In addition to the regular SIC file support, ConfigurationDesk now supports SIC files that were created with earlier Model Interface Package for Simulink/MATLAB versions. You can use the earlier SIC file versions for all platforms supported by ConfigurationDesk. The following table shows the additionally supported combinations of Model Interface Package for Simulink versions and MATLAB Releases:

Model Interface Package for Simulink Version	MATLAB Release
3.4 (dSPACE Release 2017-A)	R2015b
3.5 (dSPACE Release 2017-B)	R2016a

Model Interface Package for Simulink Version	MATLAB Release
3.6 (dSPACE Release 2018-A)	R2016b
4.0 (dSPACE Release 2018-B)	R2017a
4.1 (dSPACE Release 2019-A)	R2017b

You must observe some restrictions on the use of these SIC file versions. For more information, refer to Compatibility of ConfigurationDesk 6.7 (ConfigurationDesk Real-Time Implementation Guide 🕮).

Support of shared objects

ConfigurationDesk now supports shared objects provided by SIC files, Simulink models, and BSC files. During the build process, shared objects that match the registered platform are archived in the real-time application. Once the real-time application has been downloaded, the shared objects are available on the real-time system and can be accessed during simulation.

Support of shared objects and resources provided by FMUs

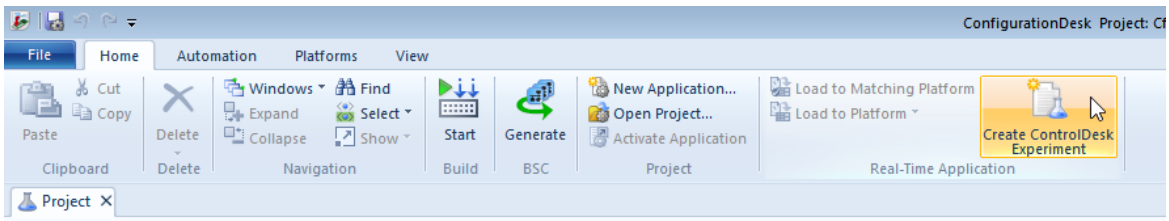
During the build process, shared objects and resources that are provided by the FMU and match the registered platform are archived in the real-time application. Once the real-time application has been downloaded, the shared objects and resources are available on the real-time system so that the FMU can access them during simulation.

Improved run-time performance for model communication of structured data ports

The run-time performance of model communication between model port blocks with structured data ports and model port blocks of other behavior models has been improved. However, the recommendations for model communications for model port blocks with structured data ports must still be observed for optimum run-time performance. Refer to Model Communication Recommendations for Model Port Blocks with Structured Data Ports (ConfigurationDesk Real-Time Implementation Guide 🕮).

Creating a ControlDesk experiment from ConfigurationDesk

You can now create a ControlDesk experiment based on the build results of the active ConfigurationDesk application by clicking Create ControlDesk Experiment on the Home ribbon:



Note

Currently this feature supports only ConfigurationDesk applications that contain a single application process, i.e., the SDF file references one variable description file. If the SDF file references multiple variable description files, e.g., in a multicore or multi-processing-unit application, the created project will be incomplete. For these applications we recommend that you still create the project and experiment in ControlDesk.

Refer to How to Create a ControlDesk Experiment from a ConfigurationDesk Application (ConfigurationDesk Real-Time Implementation Guide 🕮).

New Features Concerning I/O Functionality and Hardware Support

New function block types

The following table shows the new function block types:

Function Block Type	Desription	Supported Hardware	Channel Types	More Information
Field-Oriented Control In/Out	The **Field-Oriented Control In/Out** function block generates PWM signals to drive, for example, permanent magnet synchronous motors. The function block provides all the processing required for controlling three-phase motors via the field-oriented control method, such as transforming voltage and current vectors to space vectors and vice versa.	DS6121 (SCALEXIO)	Channels of the following channel types are requried: ▪ Digital Out 8 ▪ Analog In 16	Field-Oriented Control In/Out (ConfigurationDesk I/O Function Implementation Guide 🕮)
Lambda Probe In	The **Lambda Probe In** function block lets you control LSU 4.9 and LSU ADV wideband lambda probes.	DS1554 (MicroAutoBox III)	Channels of the following channel types are requried: ▪ Lambda In 1 ▪ Digital Out 7 or Digital In/Out 8	Lambda Probe In (ConfigurationDesk I/O Function Implementation Guide 🕮)
System Temperature Monitoring	With the **System Temperature Monitoring** function block type, you can monitor the internal temperature of MicroAutoBox III via a real-time application.	DS1403 (MicroAutoBox III)	-	System Temperature Monitoring (ConfigurationDesk I/O Function Implementation Guide 🕮)

Function Block Type	Desription	Supported Hardware	Channel Types	More Information
USB Eject	The **USB Eject** function block provides a trigger from the behavior model to eject (unmount) a USB mass storage device from the dSPACE system. Removing a USB device from the USB port without unmounting can result in data loss.	▪ DS1403 (MicroAutoBox III) ▪ DS6001 (SCALEXIO)	-	USB Eject (ConfigurationDesk I/O Function Implementation Guide 🕮)
Acceleration In	The **Acceleration In** function block type reads the measured acceleration and angular velocity values on three axes (x, y, z) from an onboard acceleration sensor and provides the values to the behavior model.	DS1403 (MicroAutoBox III)	Acceleration Sensor Unit 1	Acceleration In (ConfigurationDesk I/O Function Implementation Guide 🕮)
Atmospheric Pressure In	The **Atmospheric Pressure In** function block type reads the measured atmospheric pressure value from an onboard pressure sensor and provides the value to the behavior model.	DS1403 (MicroAutoBox III)	Pressure Sensor Unit 1	Atmospheric Pressure In (ConfigurationDesk I/O Function Implementation Guide 🕮)
LED Out	Each **LED Out** function block type provides access to one of the 4 user LEDs on the MicroAutoBox III. The LEDs can be controlled from the behavior model, for example, to display status information on the real-time application.	DS1403 (MicroAutoBox III)	LED Out 1	LED Out (ConfigurationDesk I/O Function Implementation Guide 🕮)

Enhanced function block types

FuSa System Monitoring The **FuSa System Monitoring** function block type now provides a further system monitor to trigger a FuSa response: The exception monitor.

The exception monitor can be used to monitor the processing in a real-time application. If a serious exception occurs, for example, an exception caused by a memory violation (also called segmentation fault), an internal bus error, or an invalid instruction, a FuSa response is triggered immediately.

For more information, refer to Configuring the Basic Functionality (FuSa System Monitoring) (ConfigurationDesk I/O Function Implementation Guide 🕮)

TCP The **TCP** function block type now supports the sending of keep-alive probes. With keep-alive probes, you can prevent a disconnection due to network inactivity or check whether a TCP connection is still active.

For more information, refer to Configuring the Basic Functionality (TCP) (ConfigurationDesk I/O Function Implementation Guide 🕮).

New channel types for existing function block types

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

Function Block	New Channel Types	Supported Hardware	More Information
SPI Master	Digital In/Out 6	DS1552 (MicroAutoBox III)	SPI Master (ConfigurationDesk I/O Function Implementation Guide 🕮)

New supported hardware

ConfigurationDesk now supports the following Ethernet controllers of processing hardware:

- SCALEXIO

  ConfigurationDesk now supports three Ethernet controllers of the Real-Time PC HPP 2.0, two 1 Gbit Ethernet controllers (Eth0_1 and Eth0_2), and one 10 Gbit Ethernet controller (Eth0_3).

  If you create a hardware topology from scratch, only one Ethernet controller is available. To use all Ethernet controllers, import the predefined **dSPACE SCALEXIO Processing Unit with Intel 6208U processor** HTFX file. In addition, the HTFX file lets you configure more than four processor cores of the real-time PC offline. The Real-Time PC HPP 2.0 provides 16 processor cores.

  For instructions, refer to How to Import a Hardware Topology (ConfigurationDesk Real-Time Implementation Guide 🕮).

- MicroAutoBox III

  You can now activate a third Ethernet controller of the DS1403 Processor Board. The controller is latency-optimized and recommended only for fast bypassing.

  The controller can be activated via the web interface of the MicroAutoBox III. For activation instructions, refer to How to Improve ECU Interfacing (MicroAutoBox III Hardware Installation and Configuration 🕮).

New Features of the Bus Manager in ConfigurationDesk

Capturing CAN frames

The Bus Manager now lets you capture frames that are received on a CAN bus. Capturing CAN frames is independent from communication matrices. Instead, you can capture CAN frames via frame capture elements, which you can add to the **Inspection** part of a bus configuration. The frame capture elements let you capture any CAN frame that is received in a communication cluster. Additionally, you can specify filters to capture specific frames. Via the **Frame Capture Data** feature, you can access the captured frame data, e.g., to directly provide diagnostic frames and network management frames to a mapped behavior model.

For more information, refer to Capturing CAN Frames (ConfigurationDesk Bus Manager Implementation Guide 🕮).

Enhanced J1939 support

The Bus Manager now provides the following enhancements for J1939 support:

Support of dynamic network management according to J1939-81 Dynamic network management according to J1939-81 is supported. Via the **J1939 Network Management Enable** bus configuration feature, you can enable dynamic network management for a J1939 network node. If you do this, the network node can send and respond to received address claims. Depending on the setting of its `arbitrary address capable` parameter and its priority, the network node can claim a new address in case of an address conflict.

For more information, refer to Enabling and Disabling J1939 Network Management (ConfigurationDesk Bus Manager Implementation Guide 🕮).

Support of missing sending and/or receiving ECUs such as bridge ECUs J1939 communication that is specified in a DBC communication matrix is supported even if the sending and/or receiving ECU is not available in the DBC file. For example, this can happen if bridge ECUs are used to exchange J1939 communication between different communication clusters. In this case, the bridge ECU routes J1939-compliant PDUs from one communication cluster to the other by using the transport protocol address of the original sending and/or receiving ECU, which is not available in the communication matrix. In this case, the Bus Manager generates the required ECUs, names them `DS_UnknownJ1939ECU_<transport protocol address in hex>`, and assigns the affected PDUs to the related ECUs.

For more information, refer to Working with Communication Matrices (ConfigurationDesk Bus Manager Implementation Guide 🕮)

Manipulation of J1939-compliant PDUs with up to 8 bytes J1939-compliant PDUs with up to 8 bytes are supported for manipulation. You can assign such PDUs to the **Manipulation** part of a bus configuration and manipulate the PDUs and their ISignals by using bus manipulation features.

For an overview of available bus manipulation features, refer to Basics on Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮). For basic information on using bus manipulation features, refer to Basics on Bus Manipulation Features (ConfigurationDesk Bus Manager Implementation Guide 🕮).

Configurable payload length of J1939-compliant PDUs You can configure the payload length of J1939-compliant PDUs that is specified in the communcation matrix. Refer to Configurable Settings of PDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮).

Support of extended signal multiplexing

The Bus Manager now supports extended signal multiplexing for CAN communication that is specified in DBC communication matrices. In general, the value of a *multiplexer signal* can determine that a *multiplexed signal* is included in an `extended multiplexed IPDU`. With extended signal multiplexing, a multiplexed signal itself can serve as a multiplexer signal, i.e., its value can determine other multiplexed signals to be included in the extended multiplexed IPDU.

For more information, refer to Aspects of Supported CAN Bus Features (ConfigurationDesk Bus Manager Implementation Guide 🕮).

New bus configuration features

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

Bus Configuration Feature	Bus Configuration Part	Purpose	Further Information
PDU Length	Simulated ECUs	Lets you access the payload length of CAN PDUs at run time, e.g., to simulate dynamic length CAN PDUs.	Accessing the Payload Length of CAN PDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮)
SecOC Freshness Overwrite Value	Manipulation	Lets you manipulate the freshness value of secured IPDUs with a user-defined freshness value.	Overwriting the Freshness Value of Secured IPDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮)

Generating optimized variable description files (TRC files)

The Bus Manager now lets you generate an optimized set of variables into the variable description file (TRC file). In general, variables are generated into the variable description file for all function ports with enabled test automation support. By using the new **Variable description** property of each bus configuration, you can generate an optimized set of variables for function ports of the bus configuration whose test automation support is enabled. This can significantly improve run-time performance. For newly added bus configurations, the optimized set of variables is generated by default.

For more information, refer to Accessing Function Ports with Enabled Test Automation Support in Variable Description Files (ConfigurationDesk Bus Manager Implementation Guide 🕮).

General enhancements

The Bus Manager now provides the following general enhancements:

- The performance for working with communication matrices and bus configurations in the ConfigurationDesk application and for executable applications is improved. This is achieved by various enhancements. For example:
  - By discarding unsupported communication matrix elements during the import of a communication matrix. Refer to Working with Communication Matrices (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - By disabling the evaluation of Bus Manager-related conflicts by default. Refer to Basics on Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮).
  - By improving the internal data model of the Bus Manager.
- CAN frames with a payload length of 0 bytes are supported.

- The `RX-ACCEPT-CONTAINED-IPDU` AUTOSAR attribute of container IPDUs with a dynamic container layout is evaluated. The setting of this attribute determines which IPDUs are accepted as contained IPDUs. Refer to Aspects of Supported AUTOSAR Features (ConfigurationDesk Bus Manager Implementation Guide 🕮).

  Tip

  If the `RX-ACCEPT-CONTAINED-IPDU` attribute of a container IPDU is set to `ACCEPT-CONFIGURED`, the header ID of the related contained IPDUs has to be unique only for the container IPDU and not for the entire communication matrix.

- Signals whose signal length is a multiple of 8 and exceeds 64 bits are imported as UInt8[] array signals with static length.
- Initial values that are specified in communication matrices, e.g., initial ISignal values, not only apply to the **Initial value** property of applicable bus configuration function ports, but also to the **Initial substitute value** property.

New supported AUTOSAR Release

The Bus Manager now supports AUTOSAR system description files based on AUTOSAR Classic Platform Release R20-11.

Support of further functions of the Bus Custom Code interface

The Bus Manager now supports the following functions of the Bus Custom Code interface: `DsBusCustomCodePdu_GetCanChannelName`, `DsBusCustomCodePdu_GetCanFrameTriggering`, `DsBusCustomCodePdu_GetLinFrameTriggering`, `DsBusCustomCodeSecOCPduFeature_setAuthenticatorPositionOffset`, and `DsBusCustomCodeSecOCPduFeature_getAuthenticatorPositionOffset`

For more information, refer to Overview of the Handles and Their Functions (Bus Custom Code Interface Handling 🕮).

Supported Container File Versions

Supported SIC file versions

ConfigurationDesk 6.7 supports SIC file versions as listed below:

SIC Files Created With ...	MATLAB Release
dSPACE Release 2021-A: ▪ Model Interface Package for Simulink 4.5	R2021a, R2020b, R2020a, R2019b
dSPACE Release 2020-B: ▪ Model Interface Package for Simulink 4.4 ▪ TargetLink 5.1	R2020b, R2020a, R2019b, R2019a
dSPACE Release 2020-A: ▪ Model Interface Package for Simulink 4.3	R2020a, R2019b, R2019a, R2018b

SIC Files Created With ...	MATLAB Release
dSPACE Release 2019-B: ▪ Model Interface Package for Simulink 4.2 ▪ TargetLink 5.0	R2019b, R2019a, R2018b, R2018a
dSPACE Release 2019-A: ▪ Model Interface Package for Simulink 4.1	R2017b
dSPACE Release 2018-B: ▪ Model Interface Package for Simulink 4.0	R2017a
dSPACE Release 2018-A: ▪ Model Interface Package for Simulink 3.6	R2016b
dSPACE Release 2017-B: ▪ Model Interface Package for Simulink 3.5	R2016a
dSPACE Release 2017-A: ▪ Model Interface Package for Simulink 3.4	R2015b

Note

You can use only SIC files that were generated with the Model Interface Package for Simulink for the `dsrt.tlc` system target file to build real-time applications with ConfigurationDesk. SIC files generated for the `dsrt64.tlc` system target file can be used in ConfigurationDesk only to generate BSC files for VEOS running on a Linux operating system.

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 files versions The following limitations apply to Simulink behavior models underlying SIC files created with the Model Interface Package for Simulink version 3.4 … 4.1:

- 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.
  - RTI FPGA Programming Blockset
  - MotionDesk Blockset

    Real-time applications that contain such SIC files cannot be used with MotionDesk.
  - Blocks of any dSPACE Solution.
- SIC files containing compiled objects that are not compatible with the target platform are not supported. In this case, it is not possible to use compiled objects, e.g., for IP protection scenarios.

Supported BSC file versions

ConfigurationDesk 6.7 supports BSC files that were generated with the Bus Manager of the current release, i.e., BSC file version 1.10.

Supported V-ECU implementation container versions

ConfigurationDesk 6.7 supports V-ECU implementation container versions as listed below:

V-ECU Implementations Created With...	V-ECU Implementation Version
dSPACE Release 2021-A: ▪ SystemDesk 5.5	3.0
dSPACE Release 2020-B: ▪ SystemDesk 5.5 ▪ TargetLink 5.1	3.0 As of version 3.0, *VECU* is the new V-ECU implementation file format.
dSPACE Release 2020-A: ▪ SystemDesk 5.4	2.10
dSPACE Release 2019-B: ▪ SystemDesk 5.4 ▪ TargetLink 5.0	2.10

Supported Functional Mock-up Unit versions

ConfigurationDesk 6.7 supports Functional Mock-up Units (FMUs) that comply with the following versions of the FMI standard:

- 2.0
- 2.0.1

Supported EIC file versions

ConfigurationDesk 6.7 supports EIC file versions as listed below:

EIC Files Created With ...	EIC Version
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

EIC Files Created With ...	EIC Version
dSPACE Release 2016-B (ECU Interface Manager 2.0p1)	1.0.0

However, MicroAutoBox III systems only support EIC files as of version 4.0.0.

Migrating to ConfigurationDesk 6.7

Working views containing Bus Configuration function blocks

The internal data model of bus configurations has changed. Therefore, working views (CAFX files) that contain **Bus Configuration** function blocks and that were exported with ConfigurationDesk 6.6 or earlier are no longer supported. If you try to import such a CAFX file, the import is aborted and a warning message is displayed.

Project import restricted

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

Discontinuation of SCALEXIO Ethernet Solution

The SCALEXIO Ethernet Solution is discontinued as follows:

- The end-of-life date was January 31, 2021.
- Customers with a Software Maintenance Service contract who work with dSPACE Release 2018-B will be automatically migrated to the new ConfigurationDesk UDP/TCP function blocks.

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

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

Changes to serial communication

The following changes in dSPACE Release 2021-A affect serial communication.

SENT communication An invalid value at the **Sequence** port for selecting sequence definitions from the application-specific protocol now stops the transmission/receipt of SENT messages. Until Release 2020-B, invalid values were ignored and the function block used the last valid value.

SPI communication The new **Word separation** property lets you select the method for separating consecutive data words within an SPI cycle. Until Release 2020-A, you selected the method by setting the **Time between words** property to 0 s (chip select inactive between words) or to a value greater than 0 s (transmission pause between words).

For more information on the property, refer to Configuring SPI Cycles (SPI Master) (ConfigurationDesk I/O Function Implementation Guide 🕮).

Changes to I/O Ethernet communication

The following changes in dSPACE Release 2020-B affect I/O Ethernet communication.

Receiving UDP messages The Received Bytes function port of the UDP Receive function block now continuously provides the number of valid data bytes that are provided by the Data Vector function port. Until Release 2020-A, the Received Bytes function port provided the number of valid data bytes only during the model step in which the function block provides new data.

Unsupported I/O Ethernet feature A SCALEXIO real-time application can suppress Internet group management protocol (IGMP) messages for I/O Ethernet communication.

With dSPACE Release 2020-B, dSPACE introduced a Linux-based operating system for the SCALEXIO Processing Unit and DS6001 Processor Board. The Linux-based operating system does not support the suppression of IGMP messages.

FPGA custom function blocks with APU functionality

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

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

Changes to the tool automation interface that might cause code malfunctions

The changes to the SPI property Time between words might cause existing scripts using the property to malfunction. For more information, refer to New Features and Changes to the Automation Interface for Release 2021-A (ConfigurationDesk Automating Tool Handling 🕮).

ControlDesk

Where to go from here

Information in this section

New Features of ControlDesk 7.4

Where to go from here

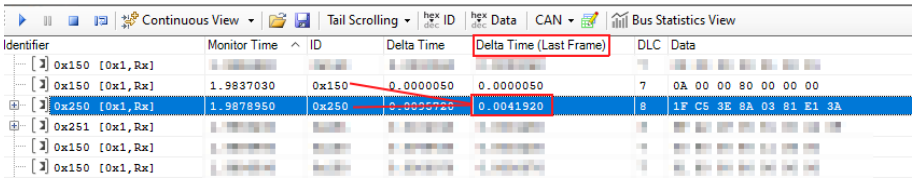
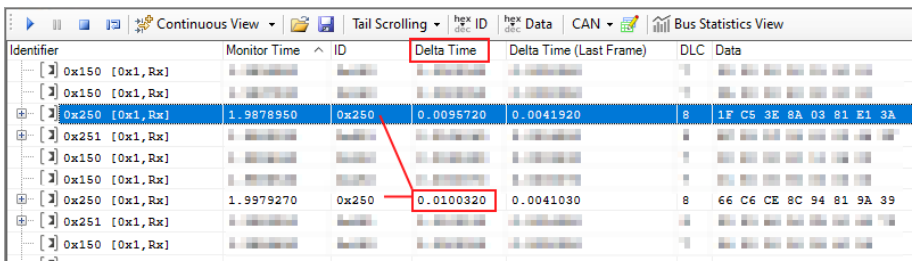
Information in this section

New Bus Navigator Features (ControlDesk 7.4)

CAN: Display of the delta time (last frame)

In addition to the *delta time*, the ControlDesk Monitoring list now lets you display the *delta time (last frame)* for CAN messages.

The following table compares the two Monitoring list columns:

Column	Description
Delta Time (Last Frame) (New in ControlDesk 7.4; available for CAN)	Displays the difference between the receipt time of a message instance and the receipt time of the previous message instance, *irrespective of the message ID* but *received by the same communication controller*. The following illustration shows an example: 
Delta Time	Displays the difference between the receipt time of a message instance and the receipt time of the previous message instance *with the same ID received by the same communication controller*. The following illustration shows an example: 

For more information, refer to Monitoring List (ControlDesk Bus Navigator 🕮).

Context menu extensibility of Bus Navigator tree nodes

ControlDesk now lets you extend the context menu of Bus Navigator tree nodes, such as communication clusters or network nodes. You can extend the context menu by using the ControlDesk automation interface.

For general information on extending the context menu of elements, refer to Extending the Context Menu of Elements (ControlDesk Customization 🕮).

New ECU Diagnostics Features (ControlDesk 7.4)

Support of CAN FD via built-in D-PDU API

ControlDesk now supports *CAN with Flexible Data Rate* (CAN FD) in connection with CAN-based ECU diagnostics via a built-in D-PDU API. The built-in D-PDU API supports all the CAN interfaces listed in Supported CAN Interfaces (ControlDesk Platform Management 🕮).

For information on the mandatory communication parameters when using a CAN FD bus, refer to Conventions in Connection with ODX Databases (ControlDesk ECU Diagnostics 🕮).

Related topics

Basics

Supported CAN Interfaces (ControlDesk Platform Management 🕮)

Migrating to ControlDesk 7.4

Where to go from here

Information in this section

Discontinuations in ControlDesk

Discontinuations as of ControlDesk 7.4

Project import supported for the last seven ControlDesk versions As of version 7.4, ControlDesk supports the direct import only of projects *last saved with one of the previous seven ControlDesk versions*.

Up to and including ControlDesk 7.3, the direct import of projects *last saved with any ControlDesk version ≥ 4.0* was supported.

Tip

To work with a ControlDesk project *last saved with a version earlier than the last seven ones*, open the project in one or more intermediate ControlDesk versions.

Support for the Calibration Hub As of ControlDesk 7.4, the Calibration Hub is no longer supported.

For new projects, you are recommended to use the DCI-CAN2 or DCI-CAN/LIN1.

Discontinuations in ControlDesk (dSPACE Release 2021-B)

Access to calculated variables via MCD-3 automation As of dSPACE Release 2021-B, the ControlDesk MCD-3 automation interface will no longer support the access to calculated variables. As a consequence, you will have to adapt MCD-3 automation scripts that involve the access to calculated variables.

Downsampling when saving displayed Time Plotter data As of dSPACE Release 2021-B, ControlDesk will no longer let you specify downsampling for saving data currently displayed in a Time Plotter to a new measurement data file.

For more information, refer to Save Displayed Data as New Measurement (ControlDesk Instrument Handling 🕮).

End of software support for discontinued dSPACE hardware

For information on the end of software support for discontinued dSPACE hardware, refer to Discontinuations on page 15.

Migrating to ControlDesk 7.4

Introduction

To migrate from ControlDesk 7.3 to ControlDesk 7.4 and reuse existing experiments, you might have to carry out the following migration steps.

Note

To migrate to ControlDesk 7.4 from versions earlier than 7.3, you also have to perform the migration steps of the intervening ControlDesk versions.

Migrating to Python 3.9

As of dSPACE Release 2021-A, dSPACE products no longer support Python 3.6. Instead, Python 3.9 is supported.

There are no ControlDesk-specific migration aspects.

For general migration aspects, refer to Migrating Python Scripts from Python 3.6 to Python 3.9 on page 27.

Message Reader API change in dSPACE Release 2021-A

There is a migration issue specific to the Message Reader API. The issue occurs if you use the API with Python. The issue was caused by the migration to Python 3.9/pythonnet 2.5.3 with dSPACE Release 2021-A.

There is no migration issue to consider if you use the API with C#.

Specifying a product filter As of dSPACE Release 2021-A, the `Products` property of the `MessageReaderSettings` class can no longer be used to set the list of products for which to filter in the log sessions. The Message Reader API provides the `SetProducts` method for this purpose.

The following table shows how to specify a product filter before and after migration:

Using Message Reader API of ...	
... dSPACE Release 2020-B and Earlier (Python 3.6)	**... dSPACE Release 2021-A and Later (Python 3.9)**

`# Specify products whose messages to read:`
`Settings = MessageReaderSettings()`
`Settings.Products.Add('ControlDesk')`
`Settings.Products.Add('AutomationDesk')`
	`# Specify products whose messages to read:`
	`Settings = MessageReaderSettings()`
	`Settings.SetProducts(['ControlDesk', 'AutomationDesk'])`

Migrating from the Vector Informatik D-PDU API to the built-in D-PDU API

As of ControlDesk 7.4, the ECU Diagnostics device supports CAN with Flexible Data Rate (CAN FD) via the built-in D-PDU API.

As a consequence, you no longer have to use the Vector Informatik D-PDU API to perform ECU diagnostics on the basis of CAN and CAN FD in connection with the ControlDesk ECU Diagnostics device. Unlike the Vector Informatik D-PDU API, the built-in D-PDU API supports not only CAN interfaces from Vector Informatik, but all the CAN interfaces listed in Supported CAN Interfaces (ControlDesk Platform Management 🕮).

Reusing experiments created on the basis of the Vector Informatik D-PDU API You do not have to perform any migration steps to reuse ECU diagnostics experiments created on the basis of the Vector Informatik D-PDU API.

Reactivating the built-in D-PDU API If you already installed and activated the Vector Informatik D-PDU API in ControlDesk 7.4 (or later) and if you want to use the built-in D-PDU API instead, you have to reactivate the latter as described below.

To activate the built-in D-PDU API again, perform the following steps:

1. Close ControlDesk, if required.
2. Replace the `D-PDU-API_RDF.xml` file that you downloaded from the Vector Informatik website and that is located in the `C:\Program Files\dSPACE ControlDesk x.y\main\bin\Diagnostic202` folder with the backup you created before you activated the Vector Informatik D-PDU API.
3. Remove the `ECUDiagnostics2_Vector.ucd` file from the ControlDesk common application settings folder.

   Tip

   You can get the path to the common application settings folder by entering the following command in the ControlDesk Interpreter controlbar:
   `Application.Environment.CommonApplicationSettingsPath`

4. Start ControlDesk again.

Migrating from earlier ControlDesk versions

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

Related topics

Basics

Basics on Migrating from Prior Versions of ControlDesk (ControlDesk Introduction and Overview 🕮)
Supported CAN Interfaces (ControlDesk Platform Management 🕮)

DCI Configuration Tool

New Features of the DCI Configuration Tool 3.13

New command line option

The command line interface of the DCI Configuration Tool now provides an option to generate ECU resets. Various parameters are available that let you control the ECU reset state.

Refer to How to Use the DCI Configuration Tool via Command Line (DCI Configuration 🕮).

Firmware version for DCI-GSI2 interfaces

For the DCI-GSI2 interfaces, the firmware version 1.5.4 is delivered with the DCI Configuration Tool 3.13.

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 FlexRay Configuration Package

New Features of dSPACE FlexRay Configuration Package 4.7

FlexRay Configuration Package

Support of AUTOSAR Classic Platform R20-11 The FlexRay Configuration Tool supports the format of AUTOSAR System Templates based on AUTOSAR Classic Platform Release R20-11 for describing FlexRay networks. However, no new features of AUTOSAR Classic Platform Release R20-11 are supported.

Refer to Communication Cluster Files Usable for Configuration (FlexRay Configuration Tool Guide 🕮).

Support of static container IPDUs The FlexRay Configuration Package now supports container IPDUs with a static container layout.

Refer to Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 🕮).

Support of cryptographic IPDUs The FlexRay Configuration Package now supports cryptographic IPDUs.

Refer to Aspects of Miscellaneous Supported AUTOSAR Features (FlexRay Configuration Tool Guide 🕮).

FlexRay Configuration Tool

Opaque byte order format The FlexRay Configuration Tool now supports signals with opaque byte order format longer than 64 bytes.

BITFIELD-TEXTTABLE computation method The FlexRay Configuration Tool now supports the BITFIELD-TEXTTABLE computation method. It is interpreted as IDENTICAL during import.

Importing an IPDU contained in several LPDUs When you import an AUTOSAR system description file, the same IPDU can be contained in multiple LPDUs without being interpreted as multiple IPDUs. This means that it is now correctly interpreted as *one* IPDU.

For FIBEX files, the correct import behavior already applied.

dSPACE Installation Manager

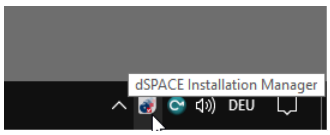
Where to go from here

Information in this section

New Features of dSPACE Installation Manager 5.7

Notification in the Windows taskbar about expiring licenses

dSPACE Installation Manager now provides a notification feature for the Windows taskbar.



If the feature is activated, the dSPACE Installation Manager automatically checks whether there is a notification for expiring licenses and displays a corresponding message on the screen if this is the case.

For more information, refer to Notifications for Expiring Licenses (Working with CodeMeter Licensing Technology 🕮).

Migrating to dSPACE Installation Manager 5.7

Using CmDongles

If you want to work with licenses on CmDongles in combination with dSPACE Installation Manager 5.7, for example, to activate, deactivate, or update licenses, the dongles must have firmware version 4.10.

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

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

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

dSPACE Python Extensions

New Features of dSPACE Python Extensions 4.0

New features

The dSPACE Python Extensions have the following new feature:

- Support of Python 3.9.

  For more information, refer to Migrating Python Scripts from Python 3.6 to Python 3.9 on page 27.

Test automation Python modules

The following features are removed from the `matlablib2` Python module:

- The MATLAB window handling is no longer supported. The following methods and properties are therefore removed:
  - `RestoreCommandWindow`
  - `MinimizeCommandWindow`
  - `MaximizeCommandWindow`
  - `Visible`
- The capture data format of ControlDesk 3.x is no longer supported. The `CaptureData` class is therefore removed.

Planned discontinuation

The `rs232lib2` Python module will be discontinued as of Release 2021-B. It is included in dSPACE Python Extensions for the last time.

As an alternative, you can use the standard Python serial port extension `pyserial`.

dSPACE XIL API .NET

New Features of dSPACE XIL API .NET 2021-A

New features

The dSPACE XIL API.NET 2021-A has the following new feature:

- Support of Python 3.9.

  For more information, refer to Migrating Python Scripts from Python 3.6 to Python 3.9 on page 27. There you find also product-specific migration aspects.

dSPACE XIL API .NET on Linux

dSPACE XIL API .NET on Linux based on dSPACE Release 2021-A is now available as a separate download package. Refer to https://www.dspace.com/go/ProductsForLinux. In contrast to the Windows support, the functionality is limited to the MAPort implementation with some restrictions.

As a new feature with Release 2021-A, platform support has been enhanced for the SCALEXIO Real-Time PC and the DS6001 Processor Board.

The user documentation of dSPACE XIL API .NET on Linux is now included in the Release documentation and therefore contained in dSPACE Help.

ECU Interface Manager

Where to go from here

Information in this section

New Features of ECU Interface Manager 2.9

Increased maximum number of ECU interface configurations that use the same interface type

Up to and including ECU Interface Manager 2.8, the number of ECU interface configurations was limited to 4 for each supported interface type, i.e., 4 for CAN and 4 for Ethernet.

As of ECU Interface Manager 2.9, the *default maximum number is 16* for each supported interface type, i.e., 16 for CAN and 16 for Ethernet. The maximum number is independent of the number of available hardware interface channels. The maximum number is now configurable.

Example Suppose you want to perform ECU interfacing via Ethernet. Using the default maximum number of 16, you can access up to 16 ECUs, for example, 10 ECUs via XCP on Ethernet and 6 ECUs via DCI-GSI2, which uses Ethernet as the transport layer.

Specifying a maximum number other than the default You can specify the maximum number in the range 1 ... 255 by using a macro for the ConfigurationDesk build process.

For more information, refer to Preconfigured Preprocessor Macros for ECU Interfacing (ConfigurationDesk Real-Time Implementation Guide 🕮).

Improved display of XCP service request messages

XCP service request messages of the SERV_TEXT type are sent by an XCP slave (ECU) when it initializes the connection to the XCP master (measurement and calibration system). If the XCP service request message display is enabled, XCP service request messages of the SERV_TEXT type are written to the dSPACE Log during run time of the real-time application. You can view them, for example, in the ControlDesk Messages controlbar.

Up to and Including ECU Interface Manager 2.8	As of ECU Interface Manager 2.9
The number of XCP service request messages of the SERV_TEXT type and buffered until being written to the dSPACE Log was limited to 10. Messages longer than 40 characters were truncated.	The *default* maximum number of XCP service request messages of the SERV_TEXT type and buffered until being written to the dSPACE Log is 50. Messages longer than 40 characters are no longer truncated.

Configuring the XCP service request message display You can configure the display of XCP service request messages of the SERV_TEXT type.

For more information, refer to Preconfigured Preprocessor Macros for ECU Interfacing (ConfigurationDesk Real-Time Implementation Guide 🕮).

Enabling/disabling priority auto-decrementation

The ECU Interface Manager uses the XCP DAQ and STIM priority mechanism to provide ECU-interfacing-related data transmission objects (DTOs) with a higher priority than DTOs used for data measurement, for example. To ensure the correct data transmission order for data consistency and event triggering purposes, the ECU Interface Manager automatically decrements the priorities of some DTOs by applying this mechanism.

However, to maintain compatibility with systems that do not support lower XCP DAQ priorities, the ECU Interface Manager now lets you disable priority auto-decrementation so that the same priority value is used on all the affected DAQ or STIM lists.

Enabling/disabling auto-decrementation of the priority value is possible for access configurations with *Manual* or *Use from Service Call* priority type.

For more information, refer to:

- Data Access (ECU Interface Manager Manual 🕮) and Function (ECU Interface Manager Manual 🕮) (access via the ECU Interface Manager user interface)
- Configure (ECU Interface Manager Manual 🕮) (access via the ECU Interface Manager automation interface)

Related topics

Basics

Preconfigured Preprocessor Macros for ECU Interfacing (ConfigurationDesk Real-Time Implementation Guide 🕮)

Compatibility of ECU Interface Manager 2.9

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 2.6[1]	Version 2.7[2]	Version 2.8[3]	Version 2.9[4]
ConfigurationDesk 6.7[4]	✓	✓	✓	✓
ConfigurationDesk 6.6[3]	✓	✓	✓	✓
ConfigurationDesk 6.5[2]	✓	✓	✓	✓
ConfigurationDesk 6.4[1]	✓	✓	✓	✓

1) dSPACE Release 2019-B
2) dSPACE Release 2020-A
3) dSPACE Release 2020-B
4) dSPACE Release 2021-A

Migrating to ECU Interface Manager 2.9

Automatic migration of projects

You can reuse projects in the ECU Interface Manager 2.9 if the projects were last saved with the ECU Interface Manager 2.0 p1 or later.

When you open the projects in the ECU Interface Manager 2.9, they are migrated automatically.

Note

In the ECU Interface Manager 2.9, you cannot reuse projects that were last saved with ECU Interface Manager 2.0 or earlier.

Additional migration steps in some cases

To migrate to the ECU Interface Manager 2.9 from versions earlier than the ECU Interface Manager 2.7, you might also have to perform the migration steps of the intervening ECU Interface Manager versions.

Firmware Manager

New Features of Firmware Manager 3.3

Enhanced platform support	The Firmware Manager supports the update of the KVM Hypervisor for the DS6001 Processor Board.

MicroAutoBox III Firmware

New Features of the MicroAutoBox III Firmware 5.1

DS1403 Processor Board

The firmware now supports the following new features:

- The functional safety monitoring is enhanced with an exception monitor. This monitor can be used to monitor the processing in a real-time application.

  For more information, refer to FuSa System Monitoring (ConfigurationDesk I/O Function Implementation Guide 🕮).
- A third Ethernet controller can be activated via the web interface. The controller is latency-optimized and recommended only for fast bypassing.

  For instructions on activating, refer to How to Improve ECU Interfacing (MicroAutoBox III Hardware Installation and Configuration 🕮).
- The acceleration and atmospheric pressure can be measured with the onboard sensors of the DS1403.

  For the characteristics of the sensors, refer to Onboard Sensor Characteristics (MicroAutoBox III Hardware Installation and Configuration 🕮).

  For implementing the sensor interfaces, refer to Acceleration In (ConfigurationDesk I/O Function Implementation Guide 🕮) and Atmospheric Pressure In (ConfigurationDesk I/O Function Implementation Guide 🕮).
- The USR LEDs can be controlled from the behavior model, for example, to display status information on the real-time application.

  For more information, refer to LED Out (ConfigurationDesk I/O Function Implementation Guide 🕮).

DS1554 Engine Control I/O Module

The firmware now supports the control of two lambda probes.

For the electrical characteristics of the lambda interface, refer to Lambda In 1 Characteristics (MicroAutoBox III Hardware Installation and Configuration 🕮).

For implementing the lambda interface, refer to Lambda Probe In (ConfigurationDesk I/O Function Implementation Guide 🕮).

Model and Sensor Interface Blockset

Where to go from here

Information in this section

New Features of Model and Sensor Interface Blockset 1.1

Model and Sensor Interface Blockset

As of dSPACE Release 2020-B, the Model and Sensor Interface Blockset (MSI Blockset) must be used for simulation with sensors in a software-in-the-loop (SIL) or in a hardware-in-the-loop (HIL) test environment.

In Sensor Simulation, the blockset is required for calculating simulation data and transmitting the data to the MotionDesk PC for visualization and to SensorSim applications that produce the sensor composition and raw data for connected sensors.

The data is transmitted in data streams of sequenced data elements using the TCP/IP protocol. In the Simulink model, the blockset must be connected to Ethernet functions to communicate with the selected supported simulation platform and to Sensor Simulation systems and applications using the TCP/IP protocol.

For each of the moving objects in the simulation, simulation data objects can be defined in kinematic chains in single object blocks using a parent-child relationship. Blocks can also be added to retrieve status and statistical information on the connected system.

You can connect an Environment Sensor Interface Unit. This splits the sensor raw data from the SensorSim application and inserts it into the relevant sensor hardware, for example, into imaging sensor interfaces of camera sensors. Additional blocks can be added to the Simulink model to receive feedback from

	the Environment Sensor Interface Unit on the connected sensors. You can also configure sensor failures on specific connected sensors, for example, a pixel error on a camera sensor. If you adapted ASM Models using the Model and Sensor Interface Solution, you can migrate these models to use the Model and Sensor Interface Blockset.
Connection Settings block transmission rate	An inport is added to the Connection Settings block to set the transmission rate during the simulation. You can select to use the inport in the Connections Settings block dialog. Therefore, you can adapt the transmission rate dynamically depending on the connected sensor without rebuilding the model. The transmission rate depends on factors such as the requested sensor update rates. The rate can be tuned to reduce the network load.
Multiple transformation data objects	The Simulation Data Objects block is updated to handle the transmission of multiple transformations, for example, to transmit the motion data for multiple fellows in a simulation. You can specify multiple objects for a single transformation object in the Simulation Data Objects block dialog. You can specify 1 - 99 objects in the dialog. Therefore, you can adapt to changes in the number of data objects, for example, fellows, without changing and rebuilding the model.
Sensor sphere support	The Simulation Data Objects block supports Position data object types. You can specify the inport dimensions for sensor contact points or sensor spheres on the objects in the scene. You can select vector or matrix dimensions.
Simulation Data Objects automation	The Simulation Data Objects automation API supports the return and setting of the number of multiple objects for a specific transformation object.
Block running order	The Model and Sensor Interface blocks are generated with a priority that defines their running order. This is to ensure synchronized data output.
Improved Simulation Data Objects dialog validation	There is improved field and parameter validation in the Simulation Data Objects dialog. Informative messages are provided if errors occur, for example, invalid characters in the parameters, name conflicts, or if other restricted actions are performed.
Improved ports interface	There is improved ports interface and signal validation.

Blockset demos

Automotive demo The blockset demo functionality includes an Automotive demo.

This demo model contains a simple ASM model used for a vehicle simulation connected to blocks of the Model and Sensor Interface Blockset. This includes a Simulation Data Objects block that contains data objects of each of the four object types: transformation, signal, position, and position and scale.

Coordinate system demo This demo model shows the influence of the transformation order on the position of the corresponding 3-D movable objects in the Model and Sensor Interface Blockset. An example for multi-object transformations has been added.

Related topics

Basics

Model and Sensor Interface Blockset Manual
New Features of MotionDesk 4.8 103
Sensor Simulation Overview

Migrating to Model and Sensor Interface Blockset 1.1

Migrating from previous Releases

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 a successful migration or if failures occur.

Models created with the Model and Sensor Interface Blockset Solution prior to dSPACE Release 2020-B must be manually migrated. When you open a model that uses blocks from the blockset solution, a messages 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 5.5

Parameterization

You can integrate a button on the parameter pages of custom libraries that starts a user-defined Python script.

Python interpreter

The interpreter has the Python version 3.9.

Migration to ModelDesk 5.5

Project migration

As of ModelDesk 5.5, you can migrate only projects created with ModelDesk version 4.5 (dSPACE Release 2017-A) 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 used in the layouts for plotting are still connected. If you replace the migrated ASM model with an ASM model created with Release 2020-B, you must reconnect the signals. ASM models created with Release 2020-B use `ASMSignalInterface` blocks that can be used in different parts of the model.

Triggering of plots

As of ModelDesk 4.6, plotting is triggered by the simulation model. Previously, ModelDesk triggered plotting. The plots are usually identical but can differ in some cases.

Tip

To compare measurements, it is useful to use the XY Plotter and use the maneuver time as a signal for the x-axis.

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 4.5

New TRC file entries for model workspace parameters of referenced models

Parameters defined in the model workspace of referenced models can now be accessed via the Model Parameters group of the referenced model. Block-level references to these parameters can be accessed via the Model Root group for the respective model. The references point to the Model Parameters group of the respective model.

Migrating to the Model Interface Package for Simulink 4.5

Unsupported new features of MATLAB R2021a

As of MATLAB R2021a, Simulink libraries can reference data dictionaries as data sources. A Simulink model that references blocks from these libraries inherits access to the referenced data dictionaries. The Model Interface Package for Simulink does not support model interfaces with data types, for example, Simulink.Bus objects, defined in the data dictionaries referenced by libraries.

Using Simulink Data Dictionaries in MATLAB R2021a

Due to a bug in MATLAB R2021a, problems can occur if you use Data Dictionaries in your Simulink model. Refer to the following bug report: https://www.mathworks.com/support/bugreports/2455953. To use Simulink Data

Dictionaries in your behavior model, it is recommended to install MATLAB R2021a Update 1 or newer.

MotionDesk

Where to go from here

Information in this section

New Features of MotionDesk 4.8

Installation

As of MotionDesk 4.7 in dSPACE Release 2020-B, MotionDesk is installed in its own folder structure in the selected installation path. It is not installed in the RCP and HIL folder.

MotionDesk Demos

In MotionDesk 4.8, the MotionDesk demos have been streamlined to support simulations using a motion data file.

For more information, refer to Migrating to MotionDesk 4.8 on page 104.

Road height map generation

Road generation is optimized to handle road networks with detailed custom height maps on roads and junctions, for example, bumpy cobblestone roads. This includes negative height maps.

In addition, lane markings, texture maps, and other random texture maps are generated on junction surfaces where height maps are used.

Tool automation for visualization management

The automation tool supports the return and setting of the motion data stream for a MotionDesk project and experiment.

The function now supports the return and setting of the status, simulation mode, and motion data stream.

Related topics

Basics

New Features of Sensor Simulation 1.5 125

Migrating to MotionDesk 4.8

Discontinuation of MotionDesk Demos

As of MotionDesk 4.8, a number of demos is discontinued.

The demos that remain available are as follows:

- ADAS Instruments
- Automotive
- Character
- City
- Scene Generation
- Sensor Simulation
- Shared Memory Access

For each of the demos, the simulation is available only using the motion data file (MDF). Demo models for supported simulation platforms are no longer delivered.

Discontinuation of the MotionDesk Blockset Demos

As of MotionDesk 4.8, the MotionDesk Blockset demos are discontinued.

The Automotive, Coorsys, and Robotic demos are discontinued and the MotionDesk Blockset demo library is no longer available. The MotionDesk demos folder is also removed from the RCP_HIL installation folder.

The MotionDesk Blockset demo `DemoMDAutomotive1` previously included in the ConfigurationDesk installation is also discontinued.

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 must use the endless sky.

Migrating 3-D custom objects	If you want to use 3-D custom objects in the VRML2 format that you used in MotionDesk 2.2.1 or earlier, you have to convert the VRML2 files to COLLADA format files. You can convert the files at any time using the 3-D Library Manager.
Migrating from 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 5.0

Int64 data type

Real-Time Testing supports the Int64 data type on the host PC.

Improved compatibility of SCALEXIO

The Python versions used on the host PC and platform must be compatible. This is always the case if the Python versions are identical. For SCALEXIO plaforms, the compatibility is extended. You can manage SCALEXIO platforms with firmware version as of 5.0 (Real-Time Testing version 4.4 included, released with Release 2020-B) from a host PC on which the current Real-Time Testing version is installed.

Python version

Real-Time Testing supports Python version 3.9 on the host PC.

Real-Time Testing on Linux

You can now run Real-time Testing on a Linux operating system.

Real-Time Testing on Linux based on dSPACE Release 2021-A is available as a separate download package. Refer to https://www.dspace.com/go/ProductsForLinux. In contrast to the Windows support, the functionality is limited:

- It supports only the following platforms: VEOS, SCALEXIO Processing Unit, and DS6001 Processor Board

- The RTT Manager is not available on Linux. You must manage the RTT sequences via Python scripts.
- The documentation of Real-Time Testing on Linux is not part of dSPACE Release 2021-A. It is provided separately in PDF format, so you can read it on a host PC with a Linux operating system.

Migrating to Real-Time Testing 5.0

Platform support

As of Real-Time Testing 4.3, the DS1005 PPC Board is no longer supported.

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 for managing real-time tests.

The BCG files generated with Real-Time Testing 4.0 or earlier cannot be used for Real-Time Testing 5.0. You must create the BCG file of the RTT sequence again.

Only 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.

C# demo scripts

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

Related topics

Basics

Creating and Starting RTT Sequences in Python Scripts (Real-Time Testing Guide 🕮)

RTI/RTI-MP and RTLib

Where to go from here

Information in this section

New Features of RTI/RTI-MP and RTLib

New features of RTI/RTI-MP

RTI and RTI-MP have the following new features:

- Support of MATLAB R2021a.
- New TRC file entries for model workspace parameters of referenced models

  Parameters defined in the model workspace of referenced models can now be accessed via the Model Parameters group of the referenced model. Block-level references to these parameters can be accessed via the Model Root group for the respective model. The references point to the Model Parameters group of the respective model.
- Modified support of A2L file generation, see below.

Modified support of A2L file generation

Release 2021-A introduces a new option and model parameter for generating A2L files.

You can now enable the generation of an A2L file with the following methods:

- By selecting the TRC and A2L option as the Variable description file format in Configuration Parameters - Code generation - RTI variable description file options.
- By assigning TRC and A2L to the VDFType model parameter via script.

By default, the generation of an A2L file is disabled and only the TRC file is generated.

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.

Note

Simulink.ConfigSet objects, which were saved in MAT files using dSPACE Release 2013-B or older, cannot be migrated to Release 2021-A.

RTI Bypass Blockset

Where to go from here

Information in this section

New Features of the RTI Bypass Blockset 3.16

RTI Bypass Blockset

Support of Green Hills compilers for on-target bypassing For on-target bypassing, the RTI Bypass Blockset now also supports C compilers from Green Hills Software for the ARM Cortex-R4/R5, Freescale MPC5xxx, Infineon TriCore, and Renesas V850/RH850 microcontroller families.

The Green Hills compilers must be installed separately.

Refer to ECU Interface Software: Packages and Modules (RTI Bypass Blockset Reference 🕮).

Compiler configuration in on-target bypassing models With the RTI Bypass Blockset, you can now manage compilers in on-target bypassing models and manually configure the compilers where allowed. You can add, copy, or remove compilers. You can also export compiler configuration settings and import compiler configurations.

Refer to Build Page (RTIBYPASS_SETUP_BLx for INTERNAL) (RTI Bypass Blockset Reference 🕮).

RTI Bypass Blockset MATLAB API

Support of compiler selection in on-target bypassing models The RTI Bypass Blockset MATLAB API lets you get all valid target-specific compiler configurations that are available for the selected code generator, and lets you select the compiler to be used for the target-specific build process. The RTI

Bypass Blockset MATLAB API also allows you to import compiler configurations from a compiler configuration file.

Refer to the RTI Bypass Blockset MATLAB API Reference 🕮.

Migrating to RTI Bypass Blockset 3.16

Models containing a DCI-GSI1-based interface

Since dSPACE Release 2020-A, the DCI-GSI1 has no longer been supported. As a consequence, the RTI Bypass Blockset no longer supports the following DCI-GSI1-based bypass interface types:

- DCI_GSI1
- dSPACE_on_JTAG_NEXUS
- dSPACE_on_JTAG_OCDS
- dSPACE_on_JTAG_SDI
- dSPACE_on_NBD_AUD
- dSPACE_on_NEXUS_READI
- cPATCH_on_JTAG_NEXUS
- cPATCH_on_JTAG_OCDS
- cPATCH_on_JTAG_SDI
- cPATCH_on_NBD_AUD
- cPATCH_on_NEXUS_READI

When you open a model with one or more of these DCI-GSI1-based bypass interfaces specified in the imported database files in the RTI Bypass Blockset 3.14 or later, a message informing you about the discontinued support is displayed, and the DCI-GSI1-based interfaces are removed from the Setup block. You can then continue working with the model as follows:

- If you did not select any of the listed interfaces in the model, you can continue working with the model as usual. No migration steps are required.
- If you selected one of the above interfaces, you must select a different bypass interface type offered in the Setup block and reconfigure all affected blocks accordingly.
- If you selected one of the above interfaces but no bypass interface type is offered for selection in the Setup block, you must import another database file with suitable interface definitions (`IF_DATA` entries) in the Setup block.

For more information, refer to Migrating Models for DCI-GSI1-Based Interfaces (RTI Bypass Blockset Reference 🕮).

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

The current Release contains RTI Bypass Blockset 3.16, which is compatible with earlier blockset versions 3.x and 2.x. However, there are some points to note:

- *Working with models from RTI Bypass Blockset 2.5 or earlier*

  Data management was changed 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 3.16, 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 3.16 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 3.16 (.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 3.15*

  If a Simulink model was built with RTI Bypass Blockset 2.6 up to RTI Bypass Blockset 3.15 and you open it with RTI Bypass Blockset 3.16, 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 3.15, you have to create a suitable database in the earlier RTI Bypass Blockset version by reimporting the database files (A2L files) specified in the Setup block.

RTI CAN MultiMessage Blockset

Where to go from here

Information in this section

New Features of the RTI CAN MultiMessage Blockset 5.6

Support of AUTOSAR Classic Platform R20-11

The RTI CAN MultiMessage Blockset supports the AUTOSAR System Template based on AUTOSAR Classic Platform Release R20-11 for describing CAN networks. However, no new features of AUTOSAR Classic Platform Release R20-11 are supported.

Refer to General Settings Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Bus Custom Code functions

The RTI CAN MultiMessage Blockset supports further functions of the Bus Custom Code interface to extend the user code. The following public Bus Custom Code functions can now be used with the RTI CAN MultiMessage Blockset:

- `DsBusCustomCodePdu_GetCanChannelName`
- `DsBusCustomCodePdu_GetCanFrameTriggering`
- `DsBusCustomCodeSecOCPduFeature_setAuthenticatorPositionOffset`
- `DsBusCustomCodeSecOCPduFeature_getAuthenticatorPositionOffset`

For more information on the functions, refer to Bus Custom Code Interface Handling 🕮.

RTI CAN MultiMessage Blockset Tutorial

The RTI CAN MultiMessage Blockset Tutorial provides a new lesson that covers the use of end-to-end communication protection (E2E protection) according to AUTOSAR E2E profiles. The lesson explains how to work with AUTOSAR E2E profiles using the profiles 05 and 11 as examples.

The checksum header file used in the tutorial lesson covers all E2E profiles supported by the RTI CAN MultiMessage Blockset. Therefore, you can use the file as a template in your projects.

Refer to Lesson 16 (Advanced): Working with E2E Communication Protection According to AUTOSAR E2E Profiles (RTI CAN MultiMessage Blockset Tutorial 🕮).

Migrating to RTI CAN MultiMessage Blockset 5.6

Working with models from earlier RTI CAN MultiMessage Blockset versions

To reuse a model created with an earlier RTI CAN MultiMessage Blockset version, you must update the S-functions for all the RTICANMM blocks and save the model before modifying the CAN configuration.

To create new S-functions for all the RTICANMM blocks in a model in one step, you can perform one of the following actions after opening the model:

- In the MATLAB Command Window, enter `rtimmsu_update('System', bdroot)`.

  For more information on the command and its options, enter `help rtimmsu_update` in the MATLAB Command Window.
- Select the **Create S-Function for all CAN Blocks** command from the **Options** menu of the RTICANMM GeneralSetup block.

For more information, refer to Limitations with RTICANMM (RTI CAN MultiMessage Blockset Reference 🕮).

Discontinuation of FIBEX 2.0 support

As of RTI CAN MultiMessage Blockset 5.6, FIBEX 2.0 files can no longer be specified as the database for an RTICANMM MainBlock.

Additional migration steps in some cases

To migrate to the RTI CAN MultiMessage Blockset 5.6 from versions earlier than the RTI CAN MultiMessage Blockset 5.5 and reuse existing models, you might have to carry out additional migration steps. For more migration information, refer to Migration (RTI CAN MultiMessage Blockset Reference 🕮).

RTI FPGA Programming Blockset

Where to go from here

Information in this section

New Features of the RTI FPGA Programming Blockset 3.11

Extended Xilinx® support

The RTI FPGA Programming Blockset now supports the following products and versions of the Xilinx design tools:

Xilinx Design Tools Version	MATLAB Version[1)]	Operating System
Vivado 2020.2[2)]	▪ MATLAB R2019b ▪ MATLAB R2020a ▪ MATLAB R2020b	Windows operating system that is supported by the RCP and HIL software of the current Release. For a list of supported operating systems, refer to Operating System on page 146. The listed Windows Server 2016 edition is not officially supported by Xilinx, but tested by dSPACE.

1) The Processor Interface sublibrary of the RTI FPGA Programming Blockset also supports MATLAB R2021a.
2) The Vivado HL WebPACK Editions of the Xilinx design tools also support the DS2655 (7K160) and DS6601 FPGA base boards. A separate license for the Xilinx System Generator for DSP is required for modeling FPGA applications with the RTI FPGA Programming Blockset.

Framework enhancements

The MicroAutoBox III and SCALEXIO frameworks now provide the following enhancements:

- When using the bus transfer mode, you can select the following bus data transmission methods to transmit data to the processor application:
  - The **Synchronous to Read_Req** method to transmit data that is captured synchronously to the processor task.
  - The **Free-running** method for applications where the transmission time is crucial.

  For more information, refer to How to Configure the Bus Data Transmission Method (RTI FPGA Programming Blockset Guide 🕮)
- DS6602 only: The framework for the DS6602 FPGA Base Board now lets you initialize the DDR4 RAM for offline simulation. Refer to Initializing the DDR4 RAM of the DS6602 for Offline Simulation (RTI FPGA Programming Blockset Guide 🕮).

Enhancements to the script interface

The following table shows the new script functions of the script interface.

Script Function	Purpose
AnalyzeFPGAXDATAWriteBus	To trigger the bus analysis of the Simulink bus that is connected to a **Buffer64 Out** FPGA block. This script function supports only MicroAutoBox III and SCALEXIO system models.
CopyFPGAXDATAReadBus	To copy an existing FPGA bus topology from a specified Simulink **Bus Creator** block, subsystem inport block, or subsystem outport block to the **Data** port of the **Buffer64 In** FPGA block. This script function supports only MicroAutoBox III and SCALEXIO system models.
GetFPGAXDATABusSettings	To get the bus settings of a **Buffer64 In/Buffer64 Out** FPGA block. This script function supports only MicroAutoBox III and SCALEXIO system models.
GetMCSubsystems	To get the Simulink subsystems that provide the access for the processor cores of a multicore system. This script function supports only MicroAutoBox III and SCALEXIO system models.
GetTraceSubsystems	To get the Simulink subsystems that are traceable with your experiment software.
ResetFPGAXDATABus	To clear the bus settings of a **Buffer64 In** FPGA block that use the bus transfer mode. This script function supports only MicroAutoBox III and SCALEXIO system models.
SetFPGAXDATABusSettings	To set the bus settings of a **Buffer64 In/Buffer64 Out** FPGA block. This script function supports only MicroAutoBox III and SCALEXIO system models.

Script Function	Purpose
SetMCSubsystems	To set the Simulink subsystems that provide the access for the processor cores of a multicore system. This script function supports only MicroAutoBox III and SCALEXIO system models.
SetTraceSubsystems	To set the Simulink subsystems that are traceable with your experiment software.

For more information, refer to Script Functions Supporting the FPGA Interface Sublibrary (RTI FPGA Programming Blockset Script Interface Reference 🕮).

Related topics

Basics

Migrating to the RTI FPGA Programming Blockset 3.11 119

Migrating to the RTI FPGA Programming Blockset 3.11

Introduction

There are various ways to migrate an existing model, depending on the blockset version used.

Migrating from RTI FPGA Programming Blockset 1.1 and higher to 3.11

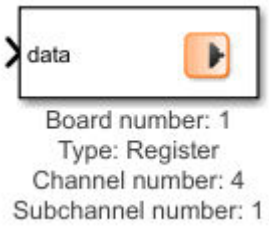
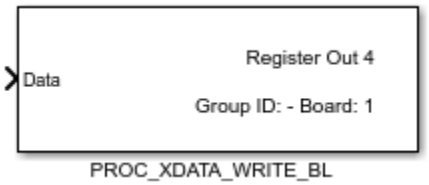
If you implemented an FPGA application with the RTI FPGA Programming Blockset Version 1.1 and later and want to use it with the RTI FPGA Programming Blockset 3.11, the framework automatically updates itself to the current framework version.

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

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 (RTI FPGA Programming Blockset Script Interface Reference 🕮).

Display of migrated processor interfaces With the RTI FPGA Programming Blockset 3.4 ... 3.8, you modeled the processor interface of a SCALEXIO system/MicroAutoBox III with processor interface blocks of the Processor Interface sublibrary.

If you migrate a model with processor interface blocks of the Processor Interface sublibrary, the update process migrates these blocks to the model port blocks of the Model Interface Package for Simulink. The following illustrations provide an example.

Display After Migration	Display Before Migration
	

ConfigurationDesk custom functions incompatible with current dSPACE Release

FPGA custom function block types that are not built with the RTI FPGA Programming Blockset 3.11 might be incompatible with the current ConfigurationDesk version.

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

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

Using different dSPACE hardware

Using an FPGA model on different dSPACE hardware requires some model modifications. Refer to Migrating to Different FPGA Hardware (RTI FPGA Programming Blockset Guide 🕮).

RTI LIN MultiMessage Blockset

Where to go from here

Information in this section

New Features of the RTI LIN MultiMessage Blockset 3.6

Support of AUTOSAR Classic Platform R20-11

The RTI LIN MultiMessage Blockset supports the AUTOSAR System Template based on AUTOSAR Classic Platform Release R20-11 for describing LIN networks. However, no new features of AUTOSAR Classic Platform Release R20-11 are supported.

Refer to General Settings Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Migrating to RTI LIN MultiMessage Blockset 3.6

Working with models from earlier RTI LIN MultiMessage Blockset versions

To reuse a model created with an earlier RTI LIN MultiMessage Blockset version, you must update the S-functions for all the RTILINMM blocks and save the model before modifying the LIN configuration.

To create new S-functions for all the RTILINMM blocks in a model in one step, you can perform one of the following actions after opening the model:

- In the MATLAB Command Window, enter `rtimmsu_update('System', bdroot)`.

For more information on the command and its options, enter `help rtimmsu_update` in the MATLAB Command Window.

- Select the **Create S-Function for all LIN Blocks** command from the **Options** menu of the RTILINMM GeneralSetup block.

For more information, refer to Limitations of RTI LIN MultiMessage Blockset (RTI LIN MultiMessage Blockset Reference 🕮).

SCALEXIO Firmware

Where to go from here

Information in this section

New Features of the SCALEXIO Firmware 5.1

Hypervisor Extension

The SCALEXIO Hypervisor Extension adds a hypervisor to the new Linux-based operating system. It can now be run on the DS6001 Processing Board.

The hypervisor is based on KVM and lets you create one or more virtual machines that run on any Linux distribution (Ubuntu 18.04 is tested and recommended).

The hypervisor provides the following advantages:

- Creation of virtual machines via configuration files.
- Parallel operation of virtual machines.
- Free distribution of memory and processing power between operating systems.
- Dynamic assignment of resources to virtual machines.
- Straightforward managing of Linux installations (as simple as with PCs).

Migrating to SCALEXIO Firmware 5.1

Migrating to the Linux-based operating system

With dSPACE Release 2020-B, dSPACE changed the SCALEXIO firmware from a QNX-based to a Linux-based distribution. The following items built for dSPACE

Release 2020-A and earlier are no longer compatible with the SCALEXIO system and must be (re-)built from source code based on dSPACE Release 2020-B or later:

- Real-time applications
- Binary libraries contained in model containers (i.e., SIC, BSC, FMU, and CTLGZ 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 older 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 memories, and the assignment of hardware resources.

Sensor Simulation

New Features of Sensor Simulation 1.5

SensorSim application

A range of fixes has been implemented in this Rrelease for Sensor Simulation. There are no new features.

Related topics

Basics

New Features of MotionDesk 4.8 103
MotionDesk Sensor Simulation Control
Sensor Simulation Manual
Sensor Simulation Overview

SYNECT

Where to go from here

Information in this section

New Features of SYNECT 2.11

Where to go from here

Information in this section

New General Features of SYNECT

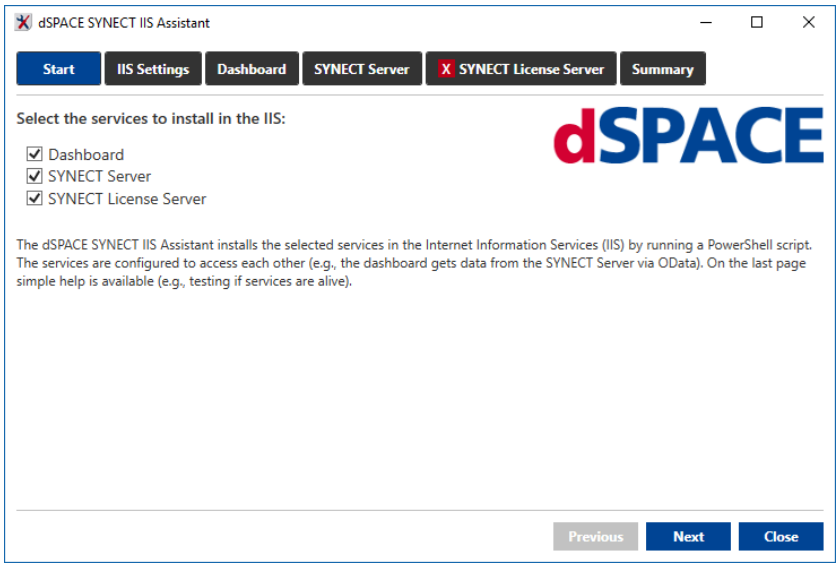
IIS assistant

SYNECT now provides an IIS assistant that simplifies hosting a SYNECT server and the related applications in Microsoft Internet Information Services (IIS).

In addition to the SYNECT server, you can host the SYNECT license server that is required to check licenses when a client requests access to data or an operation of the SYNECT server. Dashboards, which you can also host, let you present data, e.g., of tests, in browsers to monitor the status of validation processes or track performance indicators.

You can host all the applications on one server machine. However, you can also host the SYNECT server, the SYNECT license server, or dashboards on different machines.

The following illustration shows the IIS assistant, which you can access from the SYNECT Server Administrator.



Further reading Refer to Using the IIS Assistant (The SYNECT Server Guide 🕮).

OpenID Connect support

With this version, SYNECT supports OpenID Connect for logon to the SYNECT server. This improves the single sign-on (SSO) support.

SYNECT now supports the following ways of authentication:

- SYNECT user name and password: The authentication provider for the basic authentication is the SYNECT server.
- Windows authentication: The authentication provider is Microsoft Active Directory. Windows users have to be added to the SYNECT database.
- OpenID Connect: SYNECT supports OpenID Connect providers, such as Keycloak.

Further reading Refer to Basics on Authentication (The SYNECT Server Guide 🕮).

Defining state models

You can now define state models for item types. This supports operating processes along a prescribed path of state transitions.

SYNECT provides the following features for this:

- Enabling an attribute for the state model of an item type. The attribute must be an enumeration whose members define possible state values.
- Creating state transitions that can only be switched by specific user groups. This lets you define a path of state transitions according to process rules.
- Creating self-transitions. i.e., state transitions that do not change the item state. This lets you restrict all item changes in specific states to user groups that are allowed to perform the self-transition.

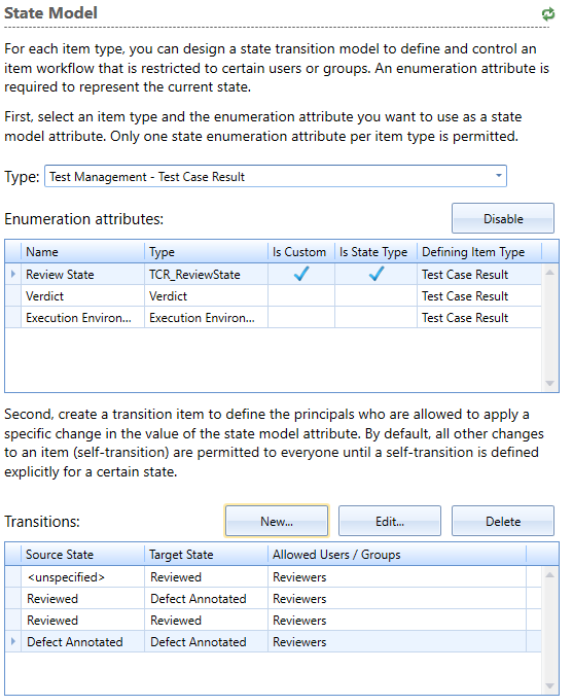
This lets you define state models for processes such as the following:

Reviewing test case results Suppose a process prescribes to review test case results. Reviewers have to annotate defects if required. The test case results must remain in a state that shows whether they were reviewed or defects were annotated. Once reviewed, items must be changed by reviewers only.

You can model the test case result review process in the following way:

- Test case results have a state attribute with Reviewed and Defect Annotated values.
- There is a group of users for the review of test case results.
- One transition allows switching test case results from the initial state to Reviewed. Another transition lets you switch the state from Reviewed to Defect Annotated. Back transitions are not defined.
- Self-transitions for the Reviewed and the Defect Annotated state values to prevent item changes of all users that are not reviewers.

The following illustration shows the state model in the backstage of SYNECT:



Further reading Refer to Basics on State Models (SYNECT Guide 🕮).

Limited support for system model containers

SYNECT model management supports the integration of system models and lets you build system models for SIL testing in VEOS. A system model contains different components that can be imported to SYNECT from container files.

With dSPACE Release 2020-B, the container format for V-ECU implementation containers changed. The new VECU format improves the data exchange between ECU software suppliers and OEMs and provides multiplatform support.

SYNECT does not support VECU containers. You cannot integrate system models with dSPACE products of dSPACE Release 2020-B or later. This affects the import of the dSPACE-specific VECU, SIC, and BSC containers as well as the exchange of system models with VEOS using SMC containers. However, the FMU 2.0 support and the support for the dSPACE-specific model containers of earlier dSPACE Releases has not changed.

Related topics

Basics

Basics on Authentication (The SYNECT Server Guide 🕮)
Basics on State Models (SYNECT Guide 🕮)
Using the IIS Assistant (The SYNECT Server Guide 🕮)

Dashboards

Presenting SYNECT data in a browser

SYNECT now provides dashboards for presenting data, such as test management data, in a browser. Dashboards let you monitor the status of tests or track key performance indicators.

You can add one or more widgets, such as the following, to dashboards:

- Data grids displaying the latest finished executions of a test management project.
- Charts presenting attributes such as the percentage of passed test cases.
- Plots showing attributes such as the verdict of test cases over time.

Refer to the following illustration, which shows a dashboard.

A dashboard application is hosted in IIS and can be opened in a browser. It provides a logon screen for authentication to the SYNECT server. The dashboard is a Java Script application that uses services for server authentication, data loading, and dashboard configuration storage.

Further reading

- For details on using dashboards to present data, such as test management data and a Getting Started example, refer to Using Dashboards (SYNECT Guide 🕮).
- For information on the application architecture of dashboards and details on hosting the application in Microsoft Internet Information Services IIS, refer to Dashboard Applications (The SYNECT Server Guide 🕮).

Related topics

Basics

Dashboard Applications (The SYNECT Server Guide 🕮)
Using Dashboards (SYNECT Guide 🕮)

New Features of Workflow Management

WFM Python API extension

With this version of SYNECT, the Workflow Management Python API has been extended. You can now set the state of the Python steps of a workflow. This lets you write workflow steps for handling the execution of workflow steps.

You have to install the **Workflow Management Python API** add-on to use and implement Python workflow steps. The API lets you access the SYNECT database to get required variant information and parameter values. You can also use the step manager to access data from the workflow step and to track step execution.

Further reading

Refer to Basics on Python Steps (SYNECT Guide 🕮).

Related topics

Basics

Basics on Python Steps (SYNECT Guide 🕮)

Migrating to SYNECT 2.11

Where to go from here

Information in this section

Migrating Databases

Introduction

To use the data from previous SYNECT versions with SYNECT 2.11, you have to migrate the SYNECT database.

To migrate databases from SYNECT Versions 2.0 - 2.10 to SYNECT 2.11, use the **Database Migrator** of SYNECT 2.11.

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 🕮).

Migrating from SYNECT 2.10

Introduction

You have to note the following points when migrating to SYNECT 2.11.

Migrating permissions

With this version of SYNECT, the **Manage User Licences** permission was introduced to improve the depth of SYNECT permissions.

In earlier SYNECT versions, the Manage Users and Groups permission included the permission to manage user licences. The permissions have been split up in this version.

Database migration specifies explicit settings for Manage Users and Groups in the following way:

SYNECT 2.10		SYNECT 2.11			
Manage Users and Groups		Manage Users and Groups		Manage User Licences	
Allow	Deny	Allow	Deny	Allow	Deny
☐	☐	☐	☐	☐	☐
☐	☑	☐	☑	☐	☑
☑	☐	☑	☐	☐ [1]	☐

[1] The permission is not automatically selected by the migration. If required, you have to specify it.

Migrating User item type queries

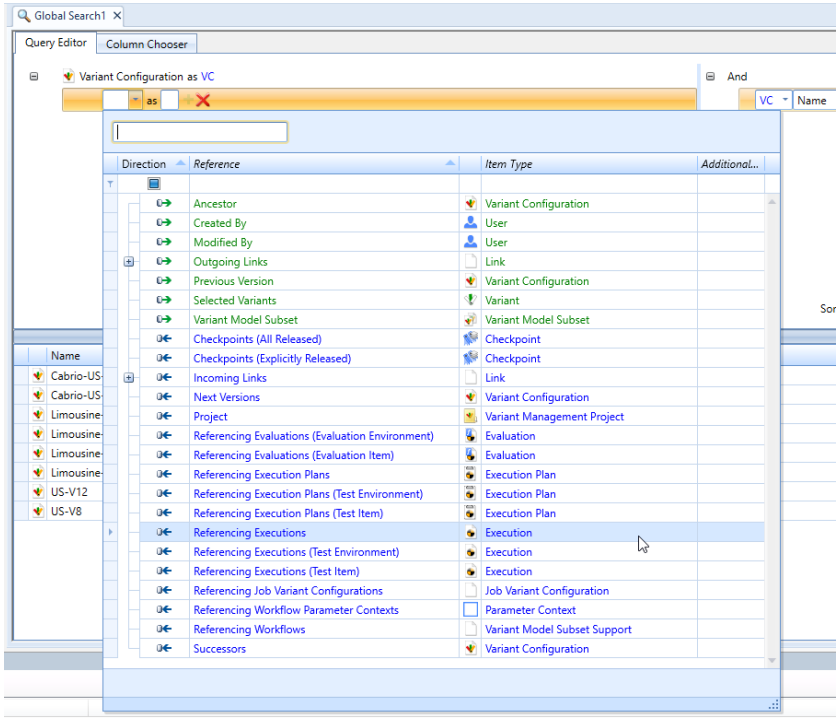
The data model of SYNECT changed with this version. You have to adapt queries that searched for User item types. Queries that used the UseWindowsAuth attribute of the item type must be changed. You can now use the Authentication Type attribute instead.

Further reading Refer to User (SYNECT Data Model Reference 🕮).

Migrating server scripts and queries that use IncomingPartTypes

The way SYNECT lets you access incoming references of item types changed. In earlier versions of SYNECT, the item type name was used. SYNECT now uses the name of the incoming reference for this.

Refer to the following illustration, which shows a query for variant configurations.



You have to adapt queries accordingly.

Old server script The old server script returns all executions that reference the variant configuration, regardless of the type of incoming reference.

```
ReturnedExecutions = MyVC.IncomingPartTypes["Variant Configuration"]
```

New server script The new script returns only the executions that reference the variant configuration via Referencing Executions

```
ReturnedExecutions = MyVC.IncomingPartTypes["Referencing Executions"]
```

You have to adapt server scripts acordingly.

Refer to the IncomingPartTypes property of ItemType (SYNECT Server API Reference 🕮).

The name of incoming references is displayed in the query editor and can be found in the following document. Refer to SYNECT Data Model Reference 🕮.

Migrating Python scripts to Python 3.9

You have to migrate Python plug-in scripts for the following reasons:

- Changed list handling of Python plug-ins
- Setting Min/Max properties of variables

Refer to Using SYNECT on page 32.

Data Model Changes from SYNECT 2.10 to SYNECT 2.11

Introduction

Some parts of the SYNECT data model have been changed from SYNECT 2.10 to SYNECT 2.11.

Deleted

Item types or reference types were not deleted.

Attributes The following attributes were deleted:

Domain	Item Type	Attributes
User Management	User (SYNECT Data Model Reference 🕮)	Use Windows Authentication

New

Item types or reference types were not added.

Attributes The following attributes were added:

Domain	Item Type	Attributes
User Management	User (SYNECT Data Model Reference 🕮)	Authentication Type

Renamed

Item types, attributes, or reference types were not renamed.

VEOS

Where to go from here

Information in this section

New Features of VEOS 5.2

Modeling enhancements

Configurable simulation step size for FMUs For FMUs, VEOS now lets you specify the simulation step size for offline simulation.

For more information, refer to Import (VEOS Manual 🕮).

Updating and replacing VPUs VEOS now lets you update and replace a VPU while keeping existing signal connections, if possible.

For more information, refer to:

- Replace (VEOS Manual 🕮)
- Update (VEOS Manual 🕮)

Support of variable-size signals and binary signals VEOS now supports the following signal types:

- Variable-size signals in BSC and VECU files
- Binary signals in VECU files

For more information, refer to Basics on Signal Connection Dependencies (VEOS Manual 🕮).

Assigning controllers to channels VEOS now lets you *assign controllers to channels*. LIN frame collisions, i.e., collisions of frames sent via two different cluster nodes, are an application example for assigning controllers to channels in VEOS.

Assigning controllers to channels in VEOS is equivalent to assigning bus access requests to bus accesses in the Bus Manager of ConfigurationDesk.

For more information, refer to Connecting Communication Controllers to Communication Clusters (VEOS Manual 🕮).

Improvement for connecting VPUs via automation The automation interface of the VEOS Player now supports a new method for connecting VPUs by using connection lists, which is useful for connecting a large number of signals.

For more information, refer to Automating the Connection of VPUs (VEOS Manual 🕮).

Running VEOS in a Linux Docker container

You can now run VEOS in a Linux Docker container.

For more information, refer to Basics on Running VEOS 5.2 in a Linux Docker Container (VEOS Manual 🕮).

Adaptive V-ECU support enhancements

GPU support for adaptive V-ECU simulation If you run VEOS on a Linux host PC equipped with a graphical processing unit (GPU), you can now use the GPU for simulating adaptive V-ECUs. VEOS 5.2 also provides a demo that shows you how to pass the GPU through to the virtual machines in which the adaptive V-ECUs are executed.

For more information, refer to Introduction to the GPU Passthrough Demo (VEOS Manual 🕮).

Support for QEMU images for adaptive V-ECU creation When you create an adaptive V-ECU for VEOS on Linux, you can now also use QEMU images as the implementation layer.

For more information, refer to Basics on Adaptive V-ECU Implementation Layers (VEOS Manual 🕮).

Command line interface extension: VEOS Support Console

The VEOS command line interface, which was introduced with VEOS 5.1, has been extended in VEOS 5.2.

The new VEOS Support Console now lets you collect support information.

For more information, refer to VEOS Support Console Command Reference (VEOS Manual 🕮).

Compatibility of VEOS 5.2

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 Manual 🕮).

Supported operating systems

Windows For information on the Windows operating systems supported by VEOS, refer to Operating System on page 146.

Linux For information on the Linux distribution recommended for VEOS, refer to Introduction to VEOS on Linux (VEOS Manual 🕮).

BSC file compatibility

VEOS 5.2 is compatible with bus simulation container (BSC) files created with the Bus Manager of dSPACE Release 2021-A (BSC version 1.10).

- If a BSC file was generated *without an SIC file*, or *with an SIC file that was generated for the dsrt.tlc system target file*, you can integrate the BSC file in a VEOS simulation on Windows.
- If a BSC file was generated *with an SIC file that was generated for the dsrt64.tlc system target file*, you can integrate the BSC file in a VEOS simulation on Linux.

CTLGZ/VECU file compatibility

The following table shows the compatibility between VEOS 5.2 and V-ECU implementation container (CTLGZ/VECU) files:

V-ECU Implementations Created With...	V-ECU Implementation Version
dSPACE Release 2021-A and 2020-B: ▪ SystemDesk 5.5 ▪ TargetLink 5.1	3.0[1)]
dSPACE Release 2020-A and 2019-B: ▪ SystemDesk 5.4 ▪ TargetLink 5.0	2.10[1)]

1) You can integrate files of this version in a VEOS simulation on Windows and in a VEOS simulation on Linux.

FMU file compatibility

- VEOS supports Functional Mock-up Units (FMUs) that comply with the FMI 2.0 standard.
- VEOS supports only the *FMI for Co-Simulation interface*, but not the FMI for Model Exchange interface.

You can integrate an FMU file in a VEOS simulation on Windows and in a VEOS simulation on Linux.

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 5.2 and offline simulation application (OSA) files:

OSA Files Created with Products Of ...	OSA Version
dSPACE Release 2021-A	5.2
dSPACE Release 2020-B	5.1[1)]
dSPACE Release 2020-A	5.0[1)]
dSPACE Release 2019-B	4.5[1)]

1) 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. Therefore, it is recommended to rebuild the binary OSA files from existing model implementation container files (CTLGZ/VECU, SIC, BSC, FMU) when you migrate from one VEOS version to another.

SIC file compatibility

The following table shows the compatibility between VEOS 5.2 and Simulink implementation container (SIC) files:

SIC Files Created With ...	SIC Version[1)]
dSPACE Release 2021-A: ▪ Model Interface Package for Simulink 4.5 ▪ TargetLink 5.1	1.10
dSPACE Release 2020-B: ▪ Model Interface Package for Simulink 4.4 ▪ TargetLink 5.1	1.9[2)]
dSPACE Release 2020-A: ▪ Model Interface Package for Simulink 4.3 ▪ TargetLink 5.0	1.8[2)]
dSPACE Release 2019-B: ▪ Model Interface Package for Simulink 4.2 ▪ TargetLink 5.0	1.7[2)]

1) If an SIC file was generated for the `dsrt.tlc` system target file, you can integrate it in a VEOS simulation on Windows. If an SIC file was generated for the `dsrt64.tlc` system target file, you can integrate it in a VEOS simulation on Linux.

SIC Files Created With ...	SIC Version[1)]

Refer to How to Generate a Simulink Implementation Container (Model Interface Package for Simulink - Modeling Guide 🕮).

[2)] 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 5.2 (dSPACE Release 2021-A). For more information, refer to Migrating ASM Models (VEOS Manual 🕮).

SMC file compatibility

The following table shows the compatibility between VEOS 5.2 and system model container (SMC) files:

SMC Files Created With ...	SMC Version
dSPACE Release 2021-A: ▪ SYNECT 2.11 ▪ VEOS 5.2	1.2
dSPACE Release 2020-B: ▪ SYNECT 2.10 ▪ VEOS 5.1	1.2
dSPACE Release 2020-A: ▪ SYNECT 2.9 ▪ VEOS 5.0	1.2
dSPACE Release 2019-B: ▪ SYNECT 2.8 ▪ VEOS 4.5	1.1

You also have to consider the following compatibility restrictions of the individual container files contained in the SMC file to be imported: If the SMC file contains a container of an unsupported version, VEOS 5.2 imports neither the unsupported container nor the connections to the application process based on the unsupported container.

Real-Time Testing compatibility

To use RTT in connection with VEOS and ControlDesk, the Real-Time Testing (RTT) version used by the VEOS Simulator that runs the simulation system and the RTT version that is active on the PC must be identical.

The following table shows the VEOS Simulator version and the corresponding RTT version:

VEOS Simulator	RTT Version
... 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
... of VEOS 4.5	Real-Time Testing Version 4.2
... of VEOS 4.4	Real-Time Testing Version 4.1
... of VEOS 4.3	Real-Time Testing Version 4.0
... of VEOS 4.2	Real-Time Testing Version 3.4

VEOS Simulator	RTT Version
... of VEOS 4.1	Real-Time Testing Version 3.3
... of VEOS 4.0	Real-Time Testing Version 3.2

ControlDesk 7.4 automatically uses the VEOS Simulator of VEOS 5.2. You can therefore use RTT in connection with VEOS and ControlDesk if RTT 5.0 is active on the PC.

AUTOSAR Adaptive Platform compatibility

For the simulation of AUTOSAR adaptive V-ECUs, VEOS 5.2 is compatible with the following release of the AUTOSAR Adaptive Platform:

- R19-11

Related topics

Basics

Hypervisor Requirements for Adaptive V-ECU Simulation (VEOS Manual 🕮)

Migrating to VEOS 5.2

Introduction

To migrate from VEOS 5.1 to VEOS 5.2, you might have to perform certain migration steps.

Note

To migrate to VEOS 5.2 from versions earlier than 5.1, you might also have to perform the migration steps of the intervening VEOS versions.

Changed naming conventions for V-ECUs

The naming conventions for V-ECUs based on the VECU container file format (V-ECU implementation version 3.0 introduced with dSPACE Release 2020-B) have been changed:

- VEOS 5.1 derived the V-ECU name *from the container file name of the related container*.
- As of Version 5.2, VEOS derives the V-ECU name *from the model identifier in the related container*.

Migrating to Python 3.9

As of dSPACE Release 2021-A, dSPACE products no longer support Python 3.6. Instead, Python 3.9 is supported.

There are no VEOS-specific migration aspects.

For general migration aspects, refer to Migrating Python Scripts from Python 3.6 to Python 3.9 on page 27.

Migrating ASM models

You cannot simulate an ASM model on VEOS if the model is contained in an OSA or SIC file that was created with a dSPACE Release earlier than the one to which your VEOS installation belongs.

To simulate an ASM model that was last saved with a dSPACE Release earlier than the dSPACE Release to which your VEOS version belongs, perform the following steps:

1. Migrate the ASM model to the dSPACE Release to which your VEOS version belongs.

   For information on migrating ASM models, refer to Migrating ASM Models (ASM User Guide 🕮).
2. Generate a Simulink implementation container (SIC) file on the basis of the ASM model by using the *Model Interface Package for Simulink*.

   For instructions, refer to Generating Simulink Implementation Containers (Model Interface Package for Simulink - Modeling Guide 🕮).
3. Import the SIC file to the VEOS Player of your VEOS version.

   For instructions, refer to How to Import Simulink Implementations (VEOS Manual 🕮).

Migrating from 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 in VEOS as of dSPACE Release 2021-B

DsIdBusIf support

As of dSPACE Release 2021-B, VEOS will no longer support V-ECUs or V-ECU implementations with a *dSPACE Idealized Bus Interface* (DsIdBusIf) module. The DsIdBusIf module is a dSPACE-specific module to specify unsupported bus protocols using an idealized bus without specific arbitration.

Use the *Signal-to-Frame Converter* (Sfc) module instead. The Sfc module is a dSPACE-specific module that lets you connect V-ECUs with modeled bus communication to environment models or other V-ECUs that use signal-based communication, such as SIC and FMU, or buses that are not supported. The bus signals can be transferred.

For more information, refer to Signal-to-Frame Converter (Sfc) (SystemDesk Manual 🕮).

Related topics

Basics

Discontinuations in VEOS (VEOS Manual 🕮)

Compatibility Information

Where to go from here

Information in this section

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 http://www.mathworks.com/support/sysreq.html.

MATLAB Release...	...Is Supported by dSPACE Release 2021-A					
	RCP and HIL Software [1), 2)]	AutomationDesk 6.5[3)]	TargetLink 5.1	Model Compare 3.1	dSPACE Python Extensions 4.0 [4)]	XIL API .NET MAPort 2021-A
R2021a	✓[5)]	✓	–	–	✓	✓
R2020b	✓	✓	✓	✓	✓	✓
R2020a	✓	✓	✓	✓	✓	✓
R2019b	✓	✓	✓	✓	✓	✓
R2019a	–	–	✓	✓[6)]	–	–

1) 'RCP and HIL software' is a generic term for a software package containing several dSPACE software products, for example, ASM, RTI, ConfigurationDesk, and ModelDesk. These software products are installed in a common folder.
2) MATLAB/Simulink Student Suite is not supported by Automotive Simulation Models (ASM).
3) The AutomationDesk MATLAB Access Library requires MATLAB.
4) matlablib2 of dSPACE Python Extensions requires MATLAB.
5) R2021a is not supported by the RTI FPGA Programming Blockset – FPGA Interface.
6) R2019a is supported by Model Compare only if at least R2019a Update 5 is used.

For up-to-date information on additional MATLAB releases that can be used in combination with dSPACE software, refer to http://www.dspace.com/go/MATLABCompatibility.

Operating System

Windows operating system on host PC

The dSPACE products of dSPACE Release 2021-A 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, Mobile, and Windows 10 S editions are not supported.

- Long-Term Servicing Branch: LTSB 2016
- Long-Term Servicing Channel: LTSC 2019
- 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 20H2 version of the Semi Annual Channel for testing.

- Windows Server 2016 Standard and Datacenter edition, each with the Desktop Experience installation option

  Only the listed editions are supported. The Windows Server 2016 Essentials, MultiPoint Premium Server editions are not supported.
- Windows Server 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.

Some limitations apply when you use dSPACE software in conjunction with features of Windows. Refer to Limitations for Using Windows Features on page 154.

Support of Windows Docker You can also run some dSPACE products in a Windows Docker container. For more details, contact dSPACE Support (www.dspace.com/go/supportrequest).

Linux operating system on host PC

The dSPACE products of dSPACE Release 2021-A with Linux compatibility support the following operating systems:

- Ubuntu 18.04 LTS with the General Availability Kernel in the Desktop, Server, and Cloud version

Some limitations apply when you use dSPACE software in conjunction with features of Linux. Refer to Limitations for Using Linux Features on page 156.

Support of Linux Docker You can also run the dSPACE products with Linux compatibility in a Linux Docker container. For more details, refer to https://www.dspace.com/go/ProductsForLinux.

Using MicroAutoBox Embedded PC as host PC

ControlDesk can also be installed on:

- MicroAutoBox Embedded PC 6th Gen. Intel® Core™ i7-6822EQ Processor, running on Microsoft® Windows® 10 IOT Enterprise, LTSB 2016, 64-bit version
- MicroAutoBox III Embedded PC, running on

  Microsoft® Windows® 10 IoT Enterprise LTSC 2019, 64-bit version

Operating system on SYNECT server

The SYNECT server supports the following operating systems:

- The same operating systems as listed above for all dSPACE products of dSPACE Release 2021-A.
- Windows Server 2012, Windows Server 2012 R2

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 dSPACE tests license servers only with Microsoft Windows operating systems in combination with protected dSPACE software.

Note

Non-Windows operating systems, such as Ubuntu Linux, are not tested. You can use them at your own risk. dSPACE does not provide support in this case.

Valid for servers with dSPACE Installation Manager dSPACE Installation Manager supports the same operating systems as the other dSPACE software products described above.

Allowing communication

Installing of additional firewall rules Additional Windows firewall rules are installed during the installation of various dSPACE software products. For example, one rule allows communication with a dSPACE expansion box, such as AutoBox. Another rule allows MotionDesk to receive motion data from a network channel. These example rules are created by the following commands:

- `netsh advfirewall firewall add rule name="dSPACE Net Service" service=any dir=in action=allow profile=any protocol=icmpv4:0, any description="Allow the dSPACE Net Service to connect to a dSPACE expansion box via network."`
- `netsh advfirewall firewall add rule name="dSPACE MotionDesk" program=<main installation path>\dSPACE MotionDesk 2021-A\MotionDesk\Bin\MotionDesk.exe" dir=in action=allow profile=any description="Allow dSPACE MotionDesk to receive motion data via network."`

Required open TCP/IP network ports If you are using third-party firewall software on your host PC, ensure that the TCP/IP communication of dSPACE software is not blocked:

- VEOS requires the following open TCP/IP network ports: 111 (TCP and UDP), 3702 (UDP), 7214 (TCP and TCP6), 7215 (TCP and UDP), 7216 (TCP), 8090 (TCP), 9923 (UDP), 15000 (UDP), 49152 ... 65535 (TCP, TCP6 and UDP)

- dSPACE Installation Manager and CodeMeter licensing software require the following open TCP/IP network ports:
  - 22350 (TCP and UDP) for communication in a LAN network (if not changed from the default setting).
  - 22352 (TCP and UDP): To access CodeMeter WebAdmin via http.
  - 22353 (TCP and UDP): To access CodeMeter WebAdmin via https.
- dSPACE Help requires an open TCP/IP network port for interprocess communication between its components. The default port number is 11000. If this port number is already being used, another free port is used automatically. The related processes can be identified via the following prefixes: `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.
- Accessing dSPACE software via Internet or network applications (e.g., Citrix, Microsoft Remote Desktop or other terminal/device servers) or to grant such access to third parties.

If you have any questions or encounter any problems, contact dSPACE Support (www.dspace.com/go/supportrequest).

Recommended virtual machine software

dSPACE tests the functionality of dSPACE software products with current VMware products and VM hardware compatibility version 10 and version 13.

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
ASM	✓	—		—
AutomationDesk	—	✓	Known limitations: ▪ Access to DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible.	—
Bus Manager	✓	—		—
ConfigurationDesk for RapidPro	✓	—		—
ConfigurationDesk	✓	—		—
Container Manager	✓	—		—
ControlDesk	—	✓	Known limitations: ▪ Access to DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible.	—
Data Dictionary Manager	✓	—		—
DCI-GSI Configuration Package	✓	—		—
dSPACE AUTOSAR Compare	✓	—		—
dSPACE Installation Manager	✓	—		—
ECU Flash Programming Tool	✓	—		—
ECU Interface Base Package	✓	—		—
ECU bypassing target compiler	✓	—		—
Failure Simulation Package	—	✓	Supported only in combination with the VEOS platform. Combinations with other platforms are not tested and therefore not released for use on VMs.	—
Firmware Archives Firmware Manager	—	✓	Known limitations: ▪ Access to DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible.	—
FlexRay Configuration Tool	✓	—		—
Model Compare	✓	—		—
ModelDesk	—	✓	Known limitations: ▪ The Traffic Object Manager cannot show custom sensor points in the preview. ▪ Plotting occasionally does not start if a start trigger is used.	—
Model Interface Package for Simulink	✓	—		—

Product	Full Support	Support with Known Limitations		No Support
MotionDesk	—	—		✓[1]
Platform API Package	—	✓	Known limitations: ▪ Access to DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible.	—
Real-Time Testing	—	✓	Known limitations: ▪ Access to DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible.	—
RTI Blocksets (Real-Time Interface)	—	✓	Known limitations: ▪ Access to DS1006 modular systems via dSPACE link boards is not possible. ▪ Access to DS1006 modular systems via Ethernet connection and slot CPU: Communication and therefore the performance is very low. ▪ Access to DS1104 R&D Controller Boards is not possible.	—
Sensor Simulation	—	—		✓[1]
SYNECT	✓	—		—
SYNECT Server	✓	—		—
SYNECT License Server	✓	—		—
SystemDesk	✓	—		—
TargetLink	✓	—		—
VEOS	✓[2]	—		—

[1] VMs do not fulfill the requirements for graphics adapters.

[2] If you want to simulate adaptive AUTOSAR V-ECUs, refer to Hypervisor Requirements for Adaptive V-ECU Simulation (VEOS Manual 🕮).

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 🕮). 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 licences, the virtual machine requires access to the dSPACE License Server. For further instructions, refer to How to Set up a Connection Between Client and Server (Working with CodeMeter Licensing Technology 🕮).

Optimal display of dSPACE Help For an optimal display of the content in dSPACE Help, you have to activate the **ClearType** setting in the VM (= default setting).

You can access this setting via the Windows Start menu (**Start – Control Panel – Appearance and Personalization – Display – Adjust ClearType text**).

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) 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 2021-A

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

Observe the following points:

- Limitations regarding run-time compatibility in the dSPACE tool chain might occur if products from different dSPACE Releases are used together.

  If dSPACE products interact directly (through automation interfaces) or indirectly (through common file types like A2L), limitations might apply. For minor limitations, refer to the relevant product documentation. The major limitations are described in the following.

  In rare cases, an additional patch must be installed for a product to achieve run-time compatibility. For more information on the patch and whether a patch is required, refer to http://www.dspace.com/go/CompPatch.
- RCP and HIL software products (of Release 2021-A) cannot be used in combination with RCP and HIL software products from earlier dSPACE Releases.

Major limitation for working with a SCALEXIO system and with MicroAutoBox III The products for working with a SCALEXIO system and with MicroAutoBox III must be compatible. This is guaranteed only for products delivered with the same dSPACE Release. Contact dSPACE for more information.

Compatibility of real-time applications loaded to a DS1006, DS1104 or MicroAutoBox II platform If a real-time application is loaded to one of these platforms with a software product of dSPACE Release 2016-B or later, software products of dSPACE Release 2016-A (and earlier) do not detect that the loaded real-time application is the same as the real-time application stored on your host PC. In this case, you cannot work with the related software product without restrictions.

This also applies if you load a real-time application with a software product of dSPACE Release 2016-A or earlier and use software products of dSPACE Release 2016-B or later, for example, for experimenting.

Combining dSPACE products from earlier Releases

For more information and notes on the combined use of different products from and with earlier Releases, refer to http://www.dspace.com/go/ds_sw_combi.

Limitations for Using Windows Features

Motivation

Some limitations apply to using dSPACE software in conjunction with features of Windows.

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 to close the dSPACE software manually before shutting down the PC.

User Account Control

It is recommended to disable the Windows User Account Control (UAC) during the installation of dSPACE software. If you cannot disable UAC, note the following Windows behavior: If UAC is enabled, the setup programs use the administrator account instead of the user account. Therefore, it is important that

the administrator account has access to the required drives, particularly the required network drives.

USB devices

If you connect dSPACE USB devices that use cables with optoisolation to the PC for the first time, there might be a message that the device driver software was not installed successfully. However, the dSPACE device will work properly later on.

Using 4K monitors

The following dSPACE software products have limitations for working with 4K monitors:

- SYNECT: If you use 4K monitors, the SYNECT graphical user interface might not be displayed properly in some cases, but this does not cause functional limitations.
- Real-Time Testing: The Real-Time Test Manager, the user interface for handling RTT sequences, does not support working with 4K monitors.
- FlexRay Configuration Package: The FlexRay Configuration Tool does not support working with 4K monitors.

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 http://www.dspace.com/faq?346 or to the Microsoft Windows documentation.

Settings in Windows for user locale and system locale must match	MATLAB reads the user locale and system locale settings that are specified in Windows operating systems. The user locale and the system locale must match. If these settings are not the same, the system might not behave as expected when working with MATLAB and dSPACE software. For instructions on checking and changing the settings, refer to https://www.mathworks.com/help/matlab/matlab_env/setting-locale-on-windows-platforms.html?s_tid=gn_loc_drop. This affects all MATLAB versions and all Windows operating systems, that are supported by dSPACE.

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 https://ubuntu.com/blog/fips-certification-ubuntu-18-04-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.

S

V

W