

dSPACE Release

# New Features and Migration

Release 2014-B – November 2014

## 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:	<a href="http://www.dspace.com">http://www.dspace.com</a>

## How to Contact dSPACE Support

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

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

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

## Software Updates and Patches

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

## Important Notice

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

© Copyright 2000 - 2014 by:

dSPACE GmbH  
Rathenaustraße 26  
33102 Paderborn  
Germany

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

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

# Contents

<b>About This Document</b>	11
<i>Document Symbols and Conventions</i> .....	11
<i>Accessing Online Help and PDF Files</i> .....	12
<b>Overview of dSPACE Release 2014-B</b>	15
<i>General Enhancements and Changes</i> .....	15
<i>64-Bit Version of RCP and HIL Software</i> .....	20
<i>Product Version Overview</i> .....	22
<i>New Product Key Features</i> .....	25
<b>Aspects of Migrating from Previous Releases</b>	33
<i>Migrating to dSPACE Release 2014-B</i> .....	33
<b>Changes to the Python 2.7 Distribution</b>	35
<i>Main Changes in Python 2.7</i> .....	35
<i>Main Changes of the dSPACE Python Distribution</i> .....	36
<i>General Information on Using Python Installations</i> .....	36
<i>Enhancements to the Standard Python 2.7 Distribution</i> .....	37
<b>AutomationDesk</b>	39
<i>Using AutomationDesk 4.1 with dSPACE Release 2014-B</i> .....	39
<b>Automotive Simulation Models (ASM)</b>	41
ASM Base InCylinder Blockset.....	42
<i>Migrating to ASM Base InCylinder Blockset 1.9.1</i> .....	42
ASM Diesel Engine Blockset.....	43
<i>New Features of ASM Diesel Engine Blockset 2.0</i> .....	43
<i>Changes in the ASM Diesel Engine Demo Model</i> .....	45
<i>Migrating to ASM Diesel Engine Blockset 2.0</i> .....	46
ASM Diesel Exhaust Blockset.....	48
<i>ASM Diesel Exhaust Blockset 2.0</i> .....	48
ASM Diesel InCylinder Blockset.....	49

<i>Changes in the ASM Diesel InCylinder Demo Model</i> .....	49
ASM Drivetrain Basic Blockset.....	50
<i>New Features of ASM Drivetrain Basic Blockset 4.0</i> .....	50
<i>Migrating to ASM Drivetrain Basic Blockset 4.0</i> .....	50
ASM Electric Components Blockset.....	52
<i>New Features of ASM Electric Components Blockset 2.7</i> .....	52
<i>Changes in the ASM Electric Components Demo Model</i> .....	52
ASM Engine Gasoline Basic Blockset.....	53
<i>New Features of ASM Engine Gasoline Basic Blockset 2.0</i> .....	53
<i>Changes in the ASM Engine Gasoline Basic Demo Model</i> .....	54
<i>Migrating to ASM Engine Gasoline Basic Blockset 2.0</i> .....	54
ASM Engine Gasoline Blockset.....	57
<i>New Features of ASM Engine Gasoline Blockset 3.0</i> .....	57
<i>Changes in the ASM Engine Gasoline Demo Model</i> .....	58
<i>Migrating to ASM Engine Gasoline Blockset 3.0</i> .....	59
ASM Gasoline InCylinder Blockset.....	62
<i>Changes in the ASM Gasoline InCylinder Demo Model</i> .....	62
<i>Migrating to ASM Gasoline InCylinder Blockset 1.9</i> .....	62
ASM Parameterization Tool.....	63
<i>New Features of the ASM Parameterization Tool 1.6.5</i> .....	63
<i>Migrating to ASM Parameterization Tool 1.6.5</i> .....	63
ASM Pneumatics Blockset.....	65
<i>New Features of ASM Pneumatics Blockset 2.0</i> .....	65
<i>Migrating to ASM Pneumatics Blockset 2.0</i> .....	65
ASM Traffic Blockset.....	66
<i>New Features of ASM Traffic Blockset 3.1</i> .....	66
<i>Migrating to ASM Traffic Blockset 3.1</i> .....	66
ASM Turbocharger Blockset.....	67
<i>New Features of ASM Turbocharger Blockset 3.0</i> .....	67
ASM Vehicle Dynamics Blockset.....	68
<i>New Features of ASM Vehicle Dynamics Blockset 3.0</i> .....	68
<i>Changes in the ASM Vehicle Dynamics Demo Model</i> .....	70
<i>Migrating to ASM Vehicle Dynamics Blockset 3.0</i> .....	70
<b>ConfigurationDesk</b> .....	73
ConfigurationDesk – Implementation.....	74

<i>New Features of ConfigurationDesk 5.2 (Implementation Version)</i> .....	74
<i>Migrating to ConfigurationDesk 5.2</i> .....	79
<b>ControlDesk Next Generation</b>	83
New Features of ControlDesk Next Generation (ControlDesk 5.3).....	84
<i>New General Features (ControlDesk 5.3)</i> .....	84
<i>New Project and Experiment Features (ControlDesk 5.3)</i> .....	85
<i>New Features of Platform Management and Platforms/Devices (ControlDesk 5.3)</i> .....	85
<i>New Variable Management Features (ControlDesk 5.3)</i> .....	86
<i>New Visualization and Instrument Features (ControlDesk 5.3)</i> .....	86
<i>New Measurement and Recording Features (ControlDesk 5.3)</i> .....	89
<i>New Bus Navigator Features (ControlDesk 5.3)</i> .....	90
<i>New Data Set Management Features (ControlDesk 5.3)</i> .....	90
<i>New ECU Diagnostics Features (ControlDesk 5.3)</i> .....	91
<i>New Signal Editor Features (ControlDesk 5.3)</i> .....	91
<i>New Automation Features (ControlDesk 5.3)</i> .....	92
Migrating to ControlDesk Next Generation (ControlDesk 5.3).....	93
<i>Migrating to ControlDesk Next Generation (ControlDesk 5.3)</i> .....	93
<b>dSPACE HIL API .NET</b>	97
<i>New Features of dSPACE HIL API .NET 1.7</i> .....	97
<b>dSPACE Python Extensions</b>	99
<i>New Features of dSPACE Python Extensions 1.7</i> .....	99
<b>dSPACE XIL API</b>	101
<i>New Features of dSPACE XIL API 2.0</i> .....	101
<i>Migrating to dSPACE XIL API 2.0</i> .....	101
<b>ECU Interface Manager</b>	103
<i>New Features of ECU Interface Manager 1.5</i> .....	103
<i>Migrating to ECU Interface Manager 1.5</i> .....	104

<b>Firmware Manager</b>	105
<i>Features of Firmware Manager 1.2</i> .....	105
<b>dSPACE FlexRay Configuration Package</b>	107
<i>New Features of dSPACE FlexRay Configuration Package</i>	
3.4.....	107
<b>Model Compare</b>	109
<i>New Features of Model Compare 2.5</i> .....	109
<i>Migration to Model Compare 2.5</i> .....	110
<b>ModelDesk</b>	113
<i>New Features of ModelDesk 4.0</i> .....	113
<b>MotionDesk</b>	115
<i>New Features of MotionDesk 3.5</i> .....	115
<i>Migrating to MotionDesk 3.5</i> .....	116
<b>Real-Time Testing</b>	117
<i>New Features of Real-Time Testing 2.4</i> .....	117
<b>RTI/RTI-MP and RTLib</b>	119
<i>New Features of RTI/RTI-MP and RTLib</i> .....	119
<i>Migration Aspects of RTI/RTI-MP and RTLib</i> .....	122
<b>RTI Bypass Blockset</b>	125
<i>New Features of the RTI Bypass Blockset 3.3</i> .....	125
<i>Migrating to RTI Bypass Blockset 3.3</i> .....	127
<b>RTI CAN Blockset</b>	129
<i>New Features of the RTI CAN Blockset 3.3</i> .....	129
<b>RTI CAN MultiMessage Blockset</b>	131
<i>New Features of the RTI CAN MultiMessage Blockset 4.0</i> ...	131
<i>Migrating to RTI CAN MultiMessage Blockset 4.0</i> .....	132

<b>RTI Electric Motor Control Blockset</b>	135
<i>Features of RTI Electric Motor Control Blockset 1.0</i> .....	135
<b>RTI Ethernet Blockset</b>	137
<i>New Features of the RTI Ethernet Blockset 1.1</i> .....	137
<b>RTI FPGA Programming Blockset</b>	139
<i>New Features of the RTI FPGA Programming Blockset 2.8</i> ....	140
<i>Migrating to RTI FPGA Programming Blockset 2.8</i> .....	141
<b>RTI LIN MultiMessage Blockset</b>	145
<i>New Features of the RTI LIN MultiMessage Blockset 2.4</i> .....	145
<i>Migrating to RTI LIN MultiMessage Blockset 2.4</i> .....	145
<b>RTI USB Flight Recorder Blockset</b>	147
<i>New Features of the RTI USB Flight Recorder Blockset 1.2</i> ....	147
<b>SCALEXIO Firmware</b>	149
<i>New Features of the SCALEXIO Firmware 3.1</i> .....	149
<b>SystemDesk</b>	151
New Features of SystemDesk 4.3.....	152
<i>New General Features</i> .....	152
<i>Modeling Software Architectures</i> .....	153
<i>Modeling Systems</i> .....	154
<i>Configuring ECUs</i> .....	154
<i>Validating Elements</i> .....	155
Migrating to SystemDesk 4.3.....	156
<i>Migrating to SystemDesk 4.3</i> .....	156
<b>TargetLink</b>	157
New Features of TargetLink 4.0 and TargetLink Data Dictionary	
4.0.....	158
Modeling in Simulink or Stateflow.....	158
<i>Support of Matrix Signals</i> .....	159
<i>Newly Supported Simulink Blocks</i> .....	159
<i>Improved Bus Support</i> .....	160

<i>Dynamic Look-Up Tables</i> .....	160
<i>Improvements to TargetLink's Simulation Frame</i> .....	161
<i>Improvements to Scaling-Invariant Systems</i> .....	162
<i>Additionally Supported Block Properties</i> .....	162
<i>Central Specification of Function Subsystem Signatures</i> .....	163
Code Generation Core Functionality.....	164
<i>MISRA-C Compliance</i> .....	165
<i>Improved Code Efficiency</i> .....	166
Data Dictionary and Data Management.....	167
<i>Improvements to the Data Dictionary</i> .....	168
<i>New DD MATLAB API Commands</i> .....	170
AUTOSAR.....	172
<i>Supported AUTOSAR Releases</i> .....	172
<i>New AUTOSAR Features</i> .....	173
Test Support.....	175
<i>Improved Online Parameter Modification</i> .....	175
<i>Changes in the Target Simulation Modules</i> .....	175
Code Generator Options.....	177
<i>New Code Generator Options</i> .....	177
Tool Chain Integration.....	179
<i>Improved Windows Compliance</i> .....	179
Other.....	180
<i>General Enhancements and Changes</i> .....	180
API Commands.....	181
<i>New API Functions</i> .....	181
Migrating to TargetLink 4.0 and TargetLink Data Dictionary 4.0 ..	184
Data Dictionary and Data Management.....	184
<i>Migrating to TargetLink 4.0</i> .....	185
<i>How to Upgrade a Data Dictionary With Included DD Files</i> ..	186
Code Changes.....	189
<i>Code Changes</i> .....	189
Code Generator Options.....	200
<i>Migration Aspects Regarding Code Generator Options</i> .....	200
Changes With Access Functions.....	201
<i>New Default Macro Bodies for Macro Access Functions</i> .....	201
<i>Changes of ADDRESS_BY_PARAMETER Access Functions</i> ....	202
<i>New Access-Function-Specific Name Macro</i> .....	203



AUTOSAR-Related Migration Aspects.....	204
<i>2-D Matrix Data Elements and Operation Arguments</i> .....	204
<i>AUTOSAR Export</i> .....	205
<i>Separate Installation of Container Manager</i> .....	205
<i>Application Data Types</i> .....	206
<i>Constraints of Array and Matrix Types</i> .....	207
<i>Replaced IsQueued Property</i> .....	207
Other.....	207
<i>Various Migration Aspects</i> .....	208
<i>Stricter Settings for Bus Diagnostics</i> .....	209
<i>Plot Channel Specification</i> .....	211
<i>Signal Properties Inheritance</i> .....	212
API Commands.....	212
<i>Changes in TargetLink and TargetLink Data Dictionary API Functions</i> .....	212
Obsolete.....	213
<i>Discontinued TargetLink Features</i> .....	213
<i>Discontinued Data Dictionary Features</i> .....	213
<i>Obsolete Limitations</i> .....	214
<i>Obsolete API Functions</i> .....	214
Messages.....	215
<i>Message Changes</i> .....	215
Stateflow-Related Changes.....	216
<i>Exported Graphical Functions</i> .....	216
<i>Stateflow Matrices</i> .....	217
Changes in Future TargetLink Versions.....	218
<i>To be Discontinued</i> .....	218
<b>VEOS</b> .....	219
<i>New Features of VEOS 3.3</i> .....	219
<i>Migrating to VEOS 3.3</i> .....	220
<b>Compatibility Information</b> .....	221
<i>Supported MATLAB Releases</i> .....	222
<i>Operating System</i> .....	224
<i>Run-Time Compatibility of dSPACE Software</i> .....	226

<i>Limitations for 64-Bit Windows Operating Systems in Combination with 32-Bit dSPACE Software.....</i>	227
<i>Limitations for Products on 64-Bit dSPACE DVD.....</i>	227
<i>Limitations for Windows 7.....</i>	230
<b>Index</b>	233

# About This Document

---

**Contents** This document informs you about the new features of all the dSPACE software products in Release 2014-B. It also gives you an overview of software products with no or minor changes. There are instructions on migrating from earlier dSPACE releases, especially from earlier product versions, if required.

---




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

<i>Document Symbols and Conventions</i>	11
<i>Accessing Online Help and PDF Files</i>	12

## Document Symbols and Conventions

---

**Symbols** The following symbols may be used in this document.

	Indicates a general hazard that may cause personal injury of any kind if you do not avoid it by following the instructions given.
	Indicates the danger of electric shock which may cause death or serious injury if you do not avoid it by following the instructions given.
	Indicates a hazard that may cause material damage if you do not avoid it by following the instructions given.



Indicates important information that should be kept in mind, for example, to avoid malfunctions.



Indicates tips containing useful information to make your work easier.


---


### Naming conventions

The following abbreviations and formats are used in this document:

**%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.

 Precedes the document title in a link that refers to another document.

 Indicates that a link refers to another document, which is available in dSPACE HelpDesk.

---

### Special folders

Some software products, for example, ControlDesk Next Generation and AutomationDesk, 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\
```

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

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

## Accessing Online Help and PDF Files

---

### Objective


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

**Online help**

You can access the online help, dSPACE HelpDesk, as follows:

**Windows Start menu** Select Start – (All) Programs – <ProductName> – dSPACE HelpDesk (<ProductName>) to open dSPACE HelpDesk with the start page of the selected product displayed. You can also navigate and search in the user documentation of any other installed software product and its supported hardware.

**Context-sensitive** Press the **F1** key or click the Help button in the dSPACE software to get help on the currently active context.

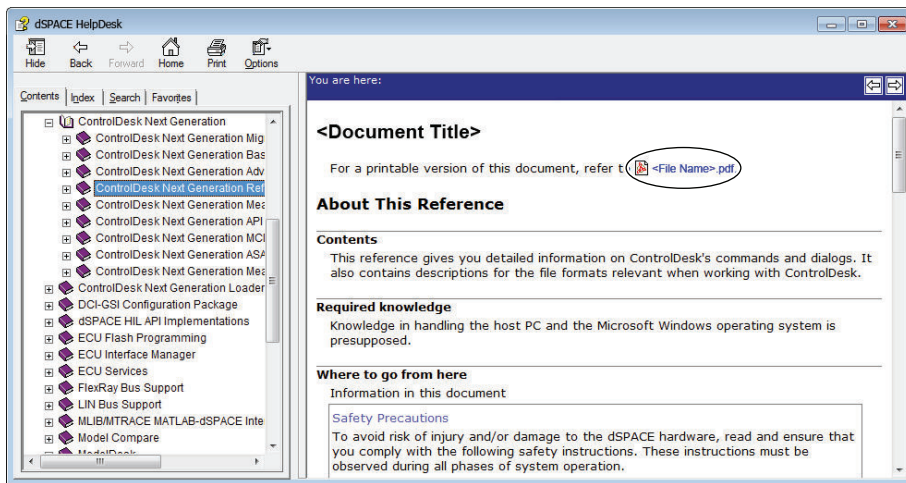
 In some software products, context-sensitive help is not available.

**Help menu in the dSPACE software** On the menu bar, select Help – Contents or Help – Search (not available in all software products) to open dSPACE HelpDesk. It opens at the start page of the currently active product. You can also navigate and search in the user documentation of any other installed software product and its supported hardware.

**PDF files**

You can access the PDF files as follows:

**dSPACE HelpDesk** Click the PDF link at the beginning of a document:





---

# Overview of dSPACE Release 2014-B

---

**Objective** Gives you an overview of the new key features in Release 2014-B and information about unchanged products.

---

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

<i>General Enhancements and Changes</i>	15
<i>64-Bit Version of RCP and HIL Software</i>	20
<i>Product Version Overview</i>	22
<i>New Product Key Features</i>	25

## General Enhancements and Changes

---

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

---

### Support of new dSPACE hardware

With dSPACE Release 2014-B, new dSPACE hardware is introduced:

- MicroLabBox

This is a new prototyping hardware for laboratory purposes. For further information, refer to *New Features of RTI/RTI-MP and RTLib* on page 119.

- DS4342 CAN FD Interface Module

This is a new piggyback module for the DS4504 Interface Board supporting the CAN FD bus protocol.

---

### Environment variables removed

As of *dSPACE RCP and HIL 2014-A*, the global environment variables that reference the installation path of this release are no longer set during installation. The dSPACE products now work independently of these environment settings.

Any custom code reading the variables `%DSPACE_ROOT%` or `%DSPACE_CONFIG%` must be migrated.

To facilitate the use of some command line tools, such as the `Down` tool for building hand-coded real-time applications, the Windows Start menu now contains a shortcut to a command prompt named *Command Prompt for dSPACE RCP and HIL <Version>*. Environment settings and search paths are then automatically set.

---

### New Python support

As of dSPACE Release 2013-B, dSPACE software products that use Python, e.g., for their automation interfaces, support Python 2.7. Some minor changes have been made in dSPACE Release 2014-B. For further information and required migration steps, refer to *Changes to the Python 2.7 Distribution* on page 35.

---

### RCP and HIL software support for MATLAB (64-bit)

The following RCP and HIL software products are now also available as product versions that support MATLAB® 64-bit versions:

- RTI Bypass Blockset

For detailed information on which products are available, refer to *64-Bit Version of RCP and HIL Software* on page 20.

---

### Distribution of 32-bit and 64-bit software

The dSPACE software is distributed on two DVD sets, each with the same content but with the following differences:

- Two 32-bit DVDs containing only 32-bit dSPACE software products, for example, to support 32-bit MATLAB versions



- Two 64-bit DVDs containing:
  - All MATLAB-related dSPACE products which have been ported to support 64-bit MATLAB versions.
  - All 32-bit dSPACE products which also support 64-bit MATLAB versions.
  - All 32-bit dSPACE products that do not relate to MATLAB (e.g., ControlDesk Next Generation).



You can therefore install dSPACE software from the 64-bit DVD set without changing to the 32-bit DVD set during the installation procedure.

For a complete list of all dSPACE products contained in dSPACE Release 2014-B as 64-bit versions, refer to *Limitations for Products on 64-Bit dSPACE DVD* on page 227.

#### Contents of DVD sets

As of dSPACE Release 2014-B, the dSPACE software is provided on two disks for each DVD set (32-bit DVD set and 64-bit DVD set). The disks contain the following dSPACE software packages and main products:

- Disk 1:
  - AutomationDesk 4.1
  - ControlDesk Next Generation (ControlDesk 5.3)
  - TargetLink 4.0
  - Model Compare 2.5



#### Product use prohibited in United States

You are not licensed to use Model Compare in the United States. You are not allowed to use or permit others to use this product in the United States or in any way that violates the laws of the United States.

- SystemDesk 4.3 (supports AUTOSAR 4.x)
  - VEOS 3.3
  - Various other dSPACE software tools
- Disk 2:
  - RCP and HIL software

"RCP and HIL software" is a generic term for a software package containing several dSPACE software products, for example RTI, ConfigurationDesk, MotionDesk, ModelDesk.

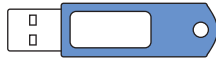


Disk 2 does not contain other dSPACE software products.

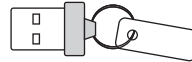
### New hardware dongles for dongle licenses

With dSPACE Release 2014-B, the hardware dongle for dongle licenses changed from WibuKey to CodeMeter. Both are products of WIBU-SYSTEMS and shown below.

WibuKey dongle



CodeMeter dongle



The new CodeMeter hardware dongles are available for new customers and are shipped for the first time with dSPACE Release 2014-B.

Keep the following compatibility information in mind:

- In general, you can use dSPACE Release 2014-B with an already delivered WibuKey dongle. With dSPACE Release 2014-B, the drivers for both dongle versions are installed on your host PC. The driver software automatically detects the dongle used. No further user action is necessary.
- If you want to use dSPACE Release 2014-A and earlier with the new CodeMeter dongle, you have to install dSPACE Installation Manager 3.8 on your host PC. This version contains the driver for the new dongle. You can download the dSPACE Installation Manager from <http://www.dspace.com/go/imupdate>.
- dSPACE Release 6.3 and earlier versions have not been tested for the new CodeMeter dongle. If necessary, contact dSPACE Support.

### Restrictions when working with dSPACE HelpDesk

dSPACE HelpDesk is installed in release-specific folders in C:\Program Files\Common Files\dSPACE on a 32-bit operating system and in C:\Program Files(x86)\Common Files\dSPACE on a 64-bit operating system. For example, if you have installed products from dSPACE Release 2014-B and products from dSPACE Release 2014-A, two dSPACE HelpDesks are available.

Note the following restrictions:

Links to documents might not work and return the following error message: *Selection is not associated with any topics.* The possible reasons are:

- The documents for the product are not installed because the product is not included in your license key.
- The documents for the product are installed in another dSPACE HelpDesk. For example, if a product in the current dSPACE Release is unchanged, its user documentation is installed in the dSPACE HelpDesk version that the product setup was created for.

After you install dSPACE Release 2014-B, you can find the user documentation for the following products in dSPACE HelpDesk 2014-A:

- AutomationDesk 4.1

If you are not sure where to find the user documentation for your product, use the product-specific dSPACE HelpDesk shortcut in the Windows Start menu to open the online help.

**Printed user documentation**

With dSPACE Release 2014-B, the printed user documentation is not delivered automatically. You can now decide which of the available printed documentation you want to have. For ordering printed documentation, refer to <http://www.dspace.com/go/requestreleasematerial>.



If you do not order printed documentation, use the online help or PDF files to obtain information about new features, enhancements, and the safety precautions regarding your products.

**Software support discontinuation**

- With dSPACE Release 2014-B, the operating system Microsoft Windows XP is no longer supported. The installation of dSPACE software is blocked.
- The following products are provided up to dSPACE Release 2014-A. As of dSPACE Release 2014-B, they have been discontinued:
  - RTI AUTOSAR Package
 

dSPACE now provides the RTI AUTOSAR Blockset 2.0 for rapid control prototyping (RCP) of V-ECUs on MicroAutoBox II.

However, the RTI AUTOSAR Blockset 2.0 is not a direct successor of the RTI AUTOSAR Package. Consequently, models created with the RTI AUTOSAR Package cannot be migrated for use with RTI AUTOSAR Blockset 2.0. Whereas the RTI AUTOSAR Package was developed to integrate single AUTOSAR software components in a Simulink model, the RTI AUTOSAR Blockset 2.0 lets you integrate V-ECU implementations in a model. In addition to atomic software components, a V-ECU implementation also contains parts of an ECU's basic software such as its operating system configuration.

RTI AUTOSAR Blockset 2.0 is not part of dSPACE Release 2014-B. To order the blockset, contact the dSPACE sales department.

- dSPACE HIL API .NET

The .NET implementation of the ASAM AE HIL API has been reduced to the MAPort implementation. The EESPort implementation has been discontinued. You can use the EESPort implementation of the new dSPACE XIL API instead of. For further information, refer to *New Features of dSPACE HIL API .NET 1.7* on page 97 and *New Features of dSPACE XIL API 2.0* on page 101.

## 64-Bit Version of RCP and HIL Software

---

### Objective

Most of the RCP and HIL software products now support 64-bit MATLAB versions.

---

### Product support in RCP and HIL (64-bit) software

In general, the RCP and HIL (64-bit) software contains the same products as the RCP and HIL software available on the dSPACE Release 2014-B (32-bit) DVD. However, not all MATLAB-related RCP and HIL software products have been ported to MATLAB x64 so far. These products are therefore not all contained in RCP and HIL (64-bit).

For an overview of RCP and HIL and all other dSPACE software products concerning the 64-bit MATLAB support, refer to *Limitations for Products on 64-Bit dSPACE DVD* on page 227.

---

**Supported MATLAB versions**

The RCP and HIL (64-bit) software supports:

- MATLAB R2013a (64-bit)
- MATLAB R2013b (64-bit)
- MATLAB R2014a (64-bit)
- MATLAB R2014b (64-bit)

See also *Supported MATLAB Releases* on page 222.

---

**Supported MEX compiler**

The RCP and HIL (64-bit) software supports only Microsoft Windows SDK 7.1 for building MEX functions.

This compiler is a free download from Microsoft. The compiler requires .NET framework 4.0, which is also available at no charge from Microsoft. Download links and instructions for the compiler and framework can be found at

<http://www.mathworks.com/support/compilers/R2012b/win64.html>.

You must install this compiler and configure it as a MEX compiler in MATLAB if you intend to use RCP and HIL products that require a MEX compiler, such as: .

- RTI CAN MultiMessage Blockset
- RTI LIN MultiMessage Blockset
- Automotive Simulation Models

---

**System requirements**

The RCP and HIL (64-bit) software requires Windows 7 Enterprise (64-bit version) with Service Pack 1. Other 64-bit operating systems (Windows XP and Windows Vista) are not supported.

The host PC main memory must be at least 4 GB RAM. 8 GB RAM or more is recommended.

See also *Operating System* on page 224.

## Product Version Overview

**Objective**

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			
	2013-A	2013-B	2014-A	2014-B
AutomationDesk	3.6p2	4.0	4.1	4.1 <sup>1)</sup>
Automotive Simulation Models	5.0	5.1	6.0	7.0 <i>See Automotive Simulation Models (ASM) on page 41.</i>
ConfigurationDesk	4.4	5.0	5.1	5.2 <i>See ConfigurationDesk on page 73.</i>
Container Manager	3.2	4.1	4.2	4.3
ControlDesk 3.x <sup>2)</sup>	3.7.5	–	–	–
ControlDesk Next Generation <sup>3)</sup>	5.0	5.1	5.2	5.3 <i>See ControlDesk Next Generation on page 83.</i>
DCI Configuration Tool	3.0	3.1	3.2.2	3.3
dSPACE CAN API	2.6	2.7	2.7.1	2.7.1
dSPACE ECU Flash Programming Tool	2.2.3	2.2.4	2.2.5	2.2.5
dSPACE FlexRay Configuration Package	3.1	3.2	3.3	3.4 <i>See dSPACE FlexRay Configuration Package on page 107.</i>
dSPACE HIL API .NET	1.4	1.5	1.6	1.7 <i>See dSPACE HIL API .NET on page 97.</i>
dSPACE Python Extensions	1.4	1.5	1.6	1.7 <i>See dSPACE Python Extensions on page 99.</i>
NEW: dSPACE XIL API	–	–	–	2.0 <i>See dSPACE XIL API on page 101.</i>

Product	dSPACE Release			
	2013-A	2013-B	2014-A	2014-B
ECU Interface Manager	1.3	1.4	1.4.1	1.5 See <i>ECU Interface Manager</i> on page 103.
Firmware Manager	–	1.0	1.1	1.2 See <i>Firmware Manager</i> on page 105.
Model Compare	2.3	2.4	2.4	2.5 See <i>Model Compare</i> on page 109.
ModelDesk	3.0	3.1	3.2	4.0 See <i>ModelDesk</i> on page 113.
MotionDesk	3.2	3.3	3.4	3.5 See <i>MotionDesk</i> on page 115.
MotionDesk Blockset	2.2.2	2.2.3	2.3	2.3.1 See <i>MotionDesk</i> on page 115.
Real-Time Testing	2.1	2.2	2.3	2.4 See <i>Real-Time Testing</i> on page 117.
RTI <sup>4)</sup>	7.0	7.1	7.2	7.3 See <i>RTI/RTI-MP and RTLib</i> on page 119.
RTI-MP <sup>5)</sup>	7.0	7.1	7.2	7.3 See <i>RTI/RTI-MP and RTLib</i> on page 119.
RTI AUTOSAR Package	1.3.1	1.3.1	1.3.1	-
RTI Bypass Blockset	3.0	3.1	3.2	3.3 See <i>RTI Bypass Blockset</i> on page 125.
RTI CAN Blockset	3.0	3.1	3.2	3.3 See <i>RTI CAN Blockset</i> on page 129.
RTI CAN MultiMessage Blockset	2.8	2.9	3.0	4.0 See <i>RTI CAN MultiMessage Blockset</i> on page 131.

Product	dSPACE Release			
	2013-A	2013-B	2014-A	2014-B
NEW: RTI Electric Motor Control Blockset	–	–	–	1.0 See <i>RTI Electric Motor Control Blockset</i> on page 135.
RTI Ethernet Blockset	–	–	1.0	1.1 See <i>RTI Ethernet Blockset</i> on page 137.
RTI Ethernet (UDP) Blockset	1.2.1	1.3	1.3	1.3
RTI FPGA Programming Blockset	2.5	2.6	2.7	2.8 See <i>RTI FPGA Programming Blockset</i> on page 139.
RTI LIN MultiMessage Blockset	2.1	2.2	2.3	2.4 See <i>RTI LIN MultiMessage Blockset</i> on page 145.
RTI RapidPro Control Unit Blockset	2.1.1	2.2	2.2	2.2
RTI USB Flight Recorder Blockset	–	1.0	1.1	1.2 See <i>RTI USB Flight Recorder Blockset</i> on page 147
RTI Watchdog Blockset	–	1.0	1.0	1.0
SCALEXIO Firmware	2.2	2.3	3.0	3.1 See <i>SCALEXIO Firmware</i> on page 149
SYNECT server	1.1	1.2	1.3.1	1.4
SystemDesk 3.x <sup>6)</sup>	3.2	3.2	–	–
SystemDesk 4.x <sup>7)</sup>	4.0	4.1	4.2	4.3 See <i>SystemDesk</i> on page 151.
TargetLink/TargetLink Data Dictionary	3.4	3.5	3.5	4.0 See <i>TargetLink</i> on page 157.



Product	dSPACE Release			
	2013-A	2013-B	2014-A	2014-B
Variable Editor	1.8	1.8	1.8	1.8
VEOS	3.0p3	3.1	3.2	3.3 See VEOS on page 219.

- <sup>1)</sup> Note the limitation described in *Using AutomationDesk 4.1 with dSPACE Release 2014-B* on page 39.
- <sup>2)</sup> ControlDesk 3.x was delivered for the last time with dSPACE Release 2013-A. This means you must migrate to ControlDesk Next Generation, the successor to ControlDesk. For migration information, refer to *Migrating from ControlDesk 3.x to ControlDesk Next Generation* ([PDF](#) *ControlDesk Next Generation Migration Guide*) and to the [PDF](#) *ControlDesk Next Generation Migration of ControlDesk 3.x Automation* document.
- <sup>3)</sup> ControlDesk Next Generation is the successor to ControlDesk 3.x and CalDesk.
- <sup>4)</sup> Including the standard I/O blocksets.
- <sup>5)</sup> Including the RTI Gigalink Blockset.
- <sup>6)</sup> Supporting AUTOSAR 3.x
- <sup>7)</sup> Supporting AUTOSAR 4.x

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

## New Product Key Features

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

**Information in this topic**

<i>AutomationDesk</i>	26
<i>ConfigurationDesk (Implementation Version)</i>	26
<i>ControlDesk Next Generation</i>	27
<i>dSPACE FlexRay Configuration Package</i>	28
<i>dSPACE HIL API .NET</i>	28
<i>dSPACE XIL API</i>	28
<i>ECU Interface Manager</i>	28
<i>Firmware Manager</i>	29
<i>Model Compare</i>	29
<i>ModelDesk</i>	29
<i>MotionDesk</i>	29
<i>Python Extensions</i>	29
<i>Real-Time Testing</i>	30
<i>RTI, RTI-MP and RTLib</i>	30
<i>RTI Bypass Blockset</i>	30
<i>RTI CAN Blockset</i>	30

<i>RTI CAN MultiMessage Blockset</i>	30
<i>NEW: RTI Electric Motor Control Blockset</i>	31
<i>RTI Ethernet Blockset</i>	31
<i>RTI FPGA Programming Blockset</i>	31
<i>RTI LIN MultiMessage Blockset</i>	31
<i>RTI USB Flight Recorder Blockset</i>	31
<i>SCALEXIO firmware</i>	31
<i>SystemDesk 4.x</i>	31
<i>TargetLink</i>	32
<i>VEOS</i>	32

**AutomationDesk**

Because AutomationDesk is not changed with dSPACE Release 2014-B, it does not support the new dSPACE platform MicroLabBox. For further limitations, refer to *Using AutomationDesk 4.1 with dSPACE Release 2014-B* on page 39.

**ConfigurationDesk  
(Implementation Version)**

The new key features of ConfigurationDesk are:

- New methods for modeling asynchronous tasks
- Methods for using an application process without a behavior model
- Support of V-ECU implementations containing one or more LIN controllers
- The SENT In function block and the Lambda DCR, Lambda NCCR function blocks have been enhanced.
- Improved user interface to make working with ConfigurationDesk applications easier and to provide convenient methods for common configuration tasks.

For details on the new features, refer to *ConfigurationDesk – Implementation* on page 74.

**ControlDesk  
Next Generation**

The new key features of ControlDesk Next Generation (ControlDesk 5.3) are:

- Platform/device enhancements:
  - Support of MicroLabBox
  - Video Capturing device: Playing data from video streams
  - Access to dSPACE real-time hardware with corrupted boot firmware
  - Support of Kvaser Leaf Light HS interface for CAN
- Project/experiment management enhancements:
  - Support of different SYNECT server versions
- Variable management enhancement:
  - Variable description compression
- Instrument and visualization enhancements:
  - New Time Plotter
  - Option to add a Python script to an instrument
- Measurement and recording enhancements:
  - Signal-wise loading of MDF 4.x data to handle large measurement files
  - Usage of sample count triggers as stop triggers for time-based rasters
- Data set management enhancement:
  - Handling writable measurements as parameters
- Bus Navigator enhancements:
  - CAN FD support for dSPACE platforms
  - Synchronization of monitoring/logging with measurements
- ECU Diagnostics enhancements:
  - Enhanced description of diagnostics variables in the Variable Browser
  - Fault Memory instrument: Display of DTC number and level
- Signal Editor enhancement:
  - Stimulation of V-ECU variables in a VEOS simulation
- Automation enhancements:
  - Access to meta information of variable descriptions
  - Handling message via automation

For details on the new features, refer to *New Features of ControlDesk Next Generation (ControlDesk 5.3)* on page 84.

---

### **dSPACE FlexRay Configuration Package**

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

- Support of FIBEX 4.1 files as database files

For details on the new feature, refer to *New Features of dSPACE FlexRay Configuration Package 3.4* on page 107.

---

### **dSPACE HIL API .NET**

The new key features of dSPACE HIL API .NET are:

- Support of MicroLabBox
- VEOS 3.3
- Stimulation for offline simulation applications with multiple environment VPUs

For details on the new features, refer to *dSPACE HIL API .NET* on page 97.

---

### **dSPACE XIL API**

dSPACE XIL API is a new product supporting the ASAM AE XIL API 2.0.1 standard that provides a generic simulator interface. It is the successor of the ASAM AE HIL API standard 1.0.2 with some modifications and enhancements.

The dSPACE implementation of the XIL API supports the test bench, the model access port (MAPort), and the electrical error simulation port (EESPort).

For details on the features, refer to *New Features of dSPACE XIL API 2.0* on page 101.

For information on migration from HIL API to XIL API, refer to *Migrating to dSPACE XIL API 2.0* on page 101.

---

### **ECU Interface Manager**

The new key features of the ECU Interface Manager are:

- Support of Renesas RH85x microcontrollers.
- Option to delete functions.
- Option to disable the execution of functions and write accesses permanently.
- Option to write back variable values to microcontroller register.
- Option to use the same service IDs for all instances.

For details on the ECU Interface Manager, refer to *New Features of ECU Interface Manager 1.5* on page 103.

---

**Firmware Manager**

The new key features of the Firmware Manager are:

- Support of MicroLabBox.
- Support of DS4342 CAN FD Interface Module mounted on a DS4505 Interface Board or MicroAutoBox II 1401/1511/1512.
- You can now repair a corrupted boot firmware via user interface.

For details on the new features, refer to *Features of Firmware Manager 1.2* on page 105.

---

**Model Compare**

The new key features of Model Compare are:

- Full-text search of hierarchy items by a name or part of a name
- Improved display of comparison results for bus-supporting blocks
- Improved display of LSB values
- Optional support for HIL model comparisons via dSPACE add-ons

For details on the new features, refer to *New Features of Model Compare 2.5* on page 109.

---

**ModelDesk**

The new key feature of ModelDesk is:

- Complete parameterization of simulation models, for example, engine models, based on measurements.

For details on the new feature, refer to *New Features of ModelDesk 4.0* on page 113.

---

**MotionDesk**

The new key features of MotionDesk are:

- New Multistate LED instrument
- New animated 3-D objects (characters and animals)

For details on the new features, refer to *New Features of MotionDesk 3.5* on page 115.

---

**Python Extensions**

The new key features of dSPACE HIL API Python Implementation for MAPort are:

- Support of MicroLabBox
- VEOS 3.3
- Stimulation for offline simulation applications with multiple environment VPUs

For details on the new feature, refer to *dSPACE Python Extensions* on page 99.

---

### Real-Time Testing

The new key feature of Real-Time Testing is:

- Support of VEOS 3.3

For details on the new feature, refer to *New Features of Real-Time Testing 2.4* on page 117.

---

### RTI, RTI-MP and RTLib

The new key features of RTI, RTI-MP and RTLib are:

- Support of MicroLabBox.
- Enhancements to MicroAutoBox.
- Support of C++ code by updated compilers.
- New default values for code generation settings.

For details on the new features, refer to *New Features of RTI/RTI-MP and RTLib* on page 119.

---

### RTI Bypass Blockset

The new key features of the RTI Bypass Blockset are:

- Support of MATLAB x64
- Labeling the ECU application's binary content.
- Improved assignment of FlexRay frames to FlexRay buffers.
- Support of XCP 1.2
- Enhancements to the RTI Bypass Blockset MATLAB API

For details on the new feature, refer to *New Features of the RTI Bypass Blockset 3.3* on page 125.

---

### RTI CAN Blockset

The new key feature of the RTI CAN Blockset is:

- Support of MicroLabBox.

For details on the new feature, refer to *New Features of the RTI CAN Blockset 3.3* on page 129.

---

### RTI CAN MultiMessage Blockset

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

- Support of MicroLabBox
- Support of the CAN FD protocol (CAN with Flexible Data Rate)
- Support of FIBEX 4.1 files as database files

For details on the new features, refer to *New Features of the RTI CAN MultiMessage Blockset 4.0* on page 131.

---

---

**NEW: RTI Electric Motor Control Blockset**

The RTI Electric Motor Control Blockset is a new blockset that provides special functions to implement controllers for state-of-the-art electric motors. It supports MicroLabBox.

For details on the new blockset, refer to *Features of RTI Electric Motor Control Blockset 1.0* on page 135.

---

**RTI Ethernet Blockset**

The new key features of the RTI Ethernet Blockset are:

- Support of MicroLabBox
- Documentation of the RTLib support.

For details on the new features, refer to *New Features of the RTI Ethernet Blockset 1.1* on page 137.

---

**RTI FPGA Programming Blockset**

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

- Extended Xilinx® software support.
- Support of MicroLabBox.
- General enhancements, such as a user-specific channel name for the I/O blocks.

For details on the new features, refer to *New Features of the RTI FPGA Programming Blockset 2.8* on page 140.

---

**RTI LIN MultiMessage Blockset**

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

- Support of FIBEX 4.1 files as database files

For details on the new feature, refer to *New Features of the RTI LIN MultiMessage Blockset 2.4* on page 145.

---

**RTI USB Flight Recorder Blockset**

The new key features of the RTI USB Flight Recorder Blockset are:

- Support of MicroLabBox
- Separate documentation for the RTLib support.

For details on the new features, refer to *New Features of the RTI USB Flight Recorder Blockset 1.2* on page 147.

---

**SCALEXIO firmware**

The new key feature of the SCALEXIO firmware is:

- Support of a new version of real-time PC.

For details on the new feature, refer to *New Features of the SCALEXIO Firmware 3.1* on page 149.

---

**SystemDesk 4.x**

The key feature of SystemDesk 4.3 is:

- Support of AUTOSAR 4.1.3, 4.1.2, 4.1.1, 4.0.3, and 4.0.2.

For details on the feature, refer to *New General Features* on page 152.

---

### TargetLink

The new key features of TargetLink are:

- 2-D signals (matrix support)
- Dynamic look-up tables
- Improved handling of bus signals (C structs)

For details on all the new features, refer to *New Features of TargetLink 4.0 and TargetLink Data Dictionary 4.0* on page 158.

For details on the TargetLink migration aspects (TargetLink, TargetLink AUTOSAR module, TargetLink Data Dictionary), refer to *Migrating to TargetLink 4.0 and TargetLink Data Dictionary 4.0* on page 184.

---

### VEOS

The new key features of VEOS are:

- Support of LIN bus simulation
- Stimulation of V-ECU variables in offline simulation

For details on the new features, refer to *VEOS* on page 219.



---

# Aspects of Migrating from Previous Releases

---

## Objective

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

## Migrating to dSPACE Release 2014-B

---

### Objective

After you install Release 2014-B, some additional steps might be necessary.

---

### Migrating from dSPACE Release 2014-A

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

---

### Migrating from dSPACE Release 2013-B or earlier

To migrate from dSPACE Release 2013-B or earlier to Release 2014-B, you also have to perform the migration steps of the intervening dSPACE Releases. All of the required migration steps can be performed with Release 2014-B installed.

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

### Previous release documents

The PDF files of previous releases are called `NewFeaturesAndMigrationxx.pdf`, where `xx` stands for the release number.

You can find the *New Features and Migration* files for previous releases here:

- In the installation folder of the current dSPACE HelpDesk, see `C:\Program Files<(x86)>\Common Files\dSPACE\HelpDesk 2014-B\Print\PreviousReleases`.
- On the dSPACE DVDs, see `\Doc\Print\PreviousReleases`.
- Download them from [www.dspace.com/go/migration](http://www.dspace.com/go/migration). Here you can also find *New Features and Migration* documents for very early releases.

---

# Changes to the Python 2.7 Distribution

---

## Objective

Gives you information on the changes in the Python distribution provided by dSPACE.

If you want to migrate from an earlier version of Python to Python 2.7, refer to the migration steps described in the *New Features and Migration* document for dSPACE Release 2013-B.

You can also find this information on the dSPACE website, refer to <http://www.dspace.com/go/Python27Migration>.

---

## Where to go from here

Information in this section

<i>Main Changes in Python 2.7</i>	35
<i>Main Changes of the dSPACE Python Distribution</i>	36
<i>General Information on Using Python Installations</i>	36
<i>Enhancements to the Standard Python 2.7 Distribution</i>	37

---

## Main Changes in Python 2.7

---

### Objective

Provides information on the changes in the Python 2.7 distribution.

### What's New documentation from the Python Software Foundation

The *What's New* document for the updated Python version is available from the Python Software Foundation:

- What's New for Python 2.7  
<http://docs.python.org/2.7/whatsnew/2.7.html>

## Main Changes of the dSPACE Python Distribution

### Objective

The Python distribution provided by dSPACE contains some dSPACE-specific changes.

### New module versions

The version of the Python Core was updated to 2.7.8 to include the latest release and benefit from bug fixes.

The version of PyWin32 was updated to 219.10, which includes the latest release 219.0 and dSPACE-specific changes.

### Components of the dSPACE Python distribution

The Python 2.7 distribution on the dSPACE DVD provides the following Python components.

Python Component	Version
Python Core	2.7.8
PyWin32	219.10
Numpy	1.7.1
Matplotlib	1.2.1
WxPython	2.9.4.0
Py2exe	0.6.9
Comtypes	0.6.2
PIL	1.1.7
Python for .NET	2.0p1

## General Information on Using Python Installations

### Objective

The following information is relevant if you want to use both Python versions on your computer.

**Using Python 2.5 and Python 2.7 in parallel**

Both Python versions can be used in parallel on your computer with the following restrictions:

- The file associations for PY and PYW files can only be set to one Python version. This is usually the latest installed Python version.
- Environment variables are used by both Python versions. Their values, for example, for `PYTHONHOME`, must be set to the Python installation you want to work with. For an overview of the environment variables set by Python, refer to <http://docs.python.org/2/using/cmdline.html>.

**Using dSPACE test automation with both Python versions in parallel**

If your test automation scripts use dSPACE Python modules distributed either via the dSPACE Python 2.5 setup or via the dSPACE Python Extensions setup available up to dSPACE Release 2013-A, and you do not want to migrate your scripts, you have to work with both Python versions.

## Enhancements to the Standard Python 2.7 Distribution

**Objective**

There are some dSPACE-specific enhancements to the standard Python 2.7. These either ensure the same behavior as before or solve known bugs. The following enhancements are available with dSPACE Release 2014-B.

**Enhancements to solve known Python bugs**

The following change has been made to solve a known bug from Python 2.7:

- Changes to the PyWin32 package from the former versions were adopted.
- The Python for .NET package was fixed to run with .NET 4.5.2.

For the latest information on bugs in Python 2.7 and their solutions, see <http://bugs.python.org>.



To identify the PyWin32 files changed by dSPACE the version number is changed from 219.0 to 219.10.



# AutomationDesk

## Using AutomationDesk 4.1 with dSPACE Release 2014-B

---

### Limitations when using AutomationDesk 4.1 with dSPACE Release 2014-B

AutomationDesk 4.1 has been released with dSPACE Release 2014-A. There is no new version on the current release DVD. There are therefore some limitations.

---

### Checking product activation

Before you start working with AutomationDesk, you should open the dSPACE Installation Manager to check which products are currently activated. The remote access via AutomationDesk is always done with the activated product version.

The following products might have an effect on AutomationDesk sequences:

- Python Extensions (containing Platform Management API)
- HIL API .NET
- Real-Time Testing
- ControlDesk Next Generation
- RCP and HIL (containing ModelDesk)

---

### Accessing platforms

There are the following limitations for handling and accessing dSPACE platforms:

- MicroLabBox
  - The new dSPACE platform is not supported.
- SCALEXIO

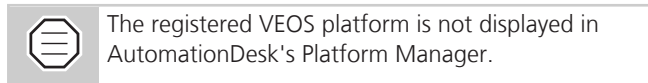
The platform management in AutomationDesk has some restrictions when using a SCALEXIO platform. You can register a SCALEXIO platform, but it is not possible to load an application and access it via the Platform Access library.

- Use the Platform Management library to load an application to the platform.
- Use the HIL API Convenience library or the HIL API library to work with the application.

#### ■ VEOS

The platform management in AutomationDesk does not work with the new VEOS version. You cannot register VEOS, and it is not possible to load an application.

- Register VEOS via ControlDesk Next Generation.



- Use the Platform Management library to load an application to the platform.

### Using demo projects

AutomationDesk demo projects with access to VEOS, SCALEXIO and Real-Time Testing use platform-specific applications. You have to replace these applications with the applications from the sample experiments installed with Real-Time Testing before you can start the AutomationDesk demo projects.

- Extract the required ZIP archive, for example, TurnSignal\_VEOS.zip, that you find in  
`C:\Program Files <(x86)>\Common Files\dSPACE\RealTimeTesting\2.4\Demos\SampleExperiments\TurnSignal_VEOS.`
- Copy the extracted folder, for example, TurnSignal\_VEOS, to AutomationDesk's TurnSignalTests\SampleExperiments demo folder.
- If required, rename the folder to the name previously used or update the paths in the AutomationDesk demo projects referencing these application files.



# Automotive Simulation Models (ASM)

## Where to go from here

### Information in this section

<i>ASM Base InCylinder Blockset</i>	42
<i>ASM Diesel Engine Blockset</i>	43
<i>ASM Diesel Exhaust Blockset</i>	48
<i>ASM Diesel InCylinder Blockset</i>	49
<i>ASM Drivetrain Basic Blockset</i>	50
<i>ASM Electric Components Blockset</i>	52
<i>ASM Engine Gasoline Basic Blockset</i>	53
<i>ASM Engine Gasoline Blockset</i>	57
<i>ASM Gasoline InCylinder Blockset</i>	62
<i>ASM Parameterization Tool</i>	63
<i>ASM Pneumatics Blockset</i>	65
<i>ASM Traffic Blockset</i>	66
<i>ASM Turbocharger Blockset</i>	67
<i>ASM Vehicle Dynamics Blockset</i>	68

### Information in other sections

#### ***Migrating ASM Models*** ( *ASM User Guide*)

Provides general information on the migration process of ASM models.

# ASM Base InCylinder Blockset

## Migrating to ASM Base InCylinder Blockset 1.9.1

---

### **EGR\_COOLER**

The block output's name has been corrected from Hdot\_Out\_ERG Cooler[J/s] to Hdot\_Out\_EGR Cooler[J/s].

# ASM Diesel Engine Blockset

## Where to go from here

Information in this section

<i>New Features of ASM Diesel Engine Blockset 2.0</i>	43
<i>Changes in the ASM Diesel Engine Demo Model</i>	45
<i>Migrating to ASM Diesel Engine Blockset 2.0</i>	46

## New Features of ASM Diesel Engine Blockset 2.0

### HPP\_CRANKBASED

The HPP\_CRANKBASED block includes a new model for a high-pressure pump of a fuel system. It calculates the volume flow through the high-pressure pump as a function of the crank angle.

The block contains two model approaches:

- A mean value model
- A pulsed model

Both approaches use the fuel delivery time as the source of volume calculation. The control signal of this pump is the start of the energization (closing) of the fuel metering unit in degrees of the crank angle.

### RAIL

The block has a new parameter: `Const_T_Rail`.

### PUMP\_TORQUE

The calculation of the compression and the implementation of the piston pressure difference have been modified.

### THROTTLE\_MECHANICAL

This is a new block for the calculation of the throttle valve position according to a control signal.

A switch for `Pos_Throttle[%]` has been added to the `MDL_In` block:

- When switching on `SoftECU`, the throttle position is calculated by the `THROTTLE_MECHANICAL` block.
- When switching on `RealECU`, the throttle position must be defined from outside the model, for example by a real throttle valve or an external model.

---

<b>THROTTLE_VALVE</b>	The block has been modified. The calculation of the throttle position has been removed. The calculation is performed in the new THROTTLE_MECHANICAL block.
<b>ENGINE_SETUP</b>	The ENGINE_SETUP block is new. It contains the basic mechanical parameters of the engine. It also contains parameters used to control the engine and set the dimensions of the signals used inside the EngineDiesel model.
<b>COMMON_ENGINE_PARAMETERS</b>	This block is new. It contains the physical constants of air, exhaust, and fuel. The COMMON_DIESEL_PARAMETERS block is discontinued. It is split up into the COMMON_ENGINE_PARAMETERS and ENGINE_SETUP blocks.
<b>RAIL_CONTROL_CRANKBASED</b>	<p>This block is new. It serves as the controller for the HPP_CRANKBASED block.</p> <p>It controls the rail pressure by calculating the fuel metering unit's (FMU's) set point of actuation. The rail pressure set point depends on the engine's operation point. A constant or an external set point can be used for testing.</p> <p>The controller is a PI controller with subsequent linear control to angle conversion. A trigger mode can be enabled to calculate the start of the FMU actuation before the working cycle and to keep that starting point until the next pump's working cycle.</p>
<b>LP_INTAKE_MANIFOLD</b>	The implementation now considers the backflow through the air path.
<b>EXHAUSTTHROTTLE</b>	The $\dot{m}_{\text{Exh}}$ and $\text{Pos\_ExhThrottle}$ inports of the look-up table parameter $\text{Map\_p\_diff\_ExhThrottle}$ have been replaced by $T_{\text{Out\_DPF}}$ and $\dot{V}_{\text{In\_ExhThrottle}}$ , because the original inports did not fit the axes of the parameter.

---

## Changes in the ASM Diesel Engine Demo Model

<b>FuelSystem</b>	<p>The FuelSystem model now contains two high-pressure pump models, one current-based and one crank angle-based model.</p> <p>The High-Pressure Pump library can be opened by clicking Open High Pressure Pump Models next to the HighPressurePump subsystem in the model. The HighPressurePump model can be replaced via drag &amp; drop of the block from the library.</p>
<b>/MDLUserInterface/Environment/MDL_PAR</b>	The signal routing has been restructured.
<b>/MDL/Environment</b>	The signal routing has been restructured.
<b>/MDL/Environment/Driver</b>	The SignalSelection block is no longer linked to an ASM library. It is now a Simulink subsystem.
<b>Cooling System</b>	The signal routing has been restructured.
<b>Throttle valve position</b>	<p>A switch for Pos_Throttle[%] has been added to the MDL_In block:</p> <ul style="list-style-type: none"> <li>■ When switching on <code>SoftECU</code>, the THROTTLE_MECHANICAL block calculates the throttle position.</li> <li>■ When switching on <code>RealECU</code>, the throttle position must be defined from outside the model, for example by a real throttle valve or an external model.</li> </ul>
<b>LP_EGR</b>	The calculation of the mass flow through the exhaust throttle has been changed. In the current version, the mass flow is calculated as the mass flow of the exhaust gas leaving the engine minus the mass flow of the high-pressure EGR and the mass flow through the low-pressure EGR cooler.

## Migrating to ASM Diesel Engine Blockset 2.0

---

**COMMON\_DIESEL\_PARAMETERS\_11\_0**

The COMMON\_DIESEL\_PARAMETERS block no longer exists. It is replaced by two new blocks: COMMON\_ENGINE\_PARAMETERS and ENGINE\_SETUP.

Its original parameters are assigned to the new blocks. The COMMON\_DIESEL\_PARAMETERS block is converted to a regular Common\_Diesel\_Parameters Simulink subsystem containing the new blocks.

---

**THROTTLE\_VALVE\_2\_0**

The THROTTLE\_VALVE block no longer exists. It is replaced by two new blocks: THROTTLE\_MECHANICAL and THROTTLE\_VALVE.

Its original parameters are assigned to the new blocks. The THROTTLE\_VALVE block is converted to a regular Simulink subsystem containing the new blocks.

---

**SWITCHES\_FUEL\_METER\_UNIT\_1\_0**

The block has been discontinued. The block's parameter has been moved to the HIGH\_PRESSURE\_PUMP block.

---

**HIGH\_PRESSURE\_PUMP**

The parameter of the discontinued SWITCHES\_FUEL\_METER\_UNIT\_1\_0 block has been moved to the HIGH\_PRESSURE\_PUMP block. In former versions, the SWITCHES\_FUEL\_METER\_UNIT parameter was an inport to the HIGH\_PRESSURE\_PUMP model. Thus, in the postmigrate variant of the HIGH\_PRESSURE\_PUMP, the setting of the new to old parameter structure takes place and the old inport is terminated in the model. The Map\_V\_dead and Const\_p\_Low parameters are added during premigration.

---

**SWITCHES\_THROTTLE\_1\_0**

The block has been discontinued.

---

**RAIL**

The Const\_T\_Rail parameter is added in the premigrate variant.

**ENGINE\_SETUP**

The `Const_max_num_Inj_Direct`, `Const_max_num_Inj_Port`, and `Const_m_Air_ref` parameters are added in the premigrate variant.

The value of the `MDL.EngineDiesel.Const.Const_max_num_Inj` parameter is assigned to the `MDL.EngineDiesel.Setup.Const_max_num_Inj_Direct` parameter.

The value of the `MDL.EngineDiesel.Const.Const_max_num_Cyl` parameter is assigned to the `MDL.EngineDiesel.Setup.Const_max_num_Cyl` parameter during postmigration.

The `Const_max_num_Inj_Direct`, `Const_max_num_Inj_Port` and `Const_max_num_Cyl` parameters are used to match the signal sizes in the engine model to the external model or the real ECU in the `MDL_In` system. Also, the ASM models from the Operator libraries will use these values to define the dimensional input to the S-function placed under their masks.

The functionality to write these values with ModelDesk is new. Therefore they have to be manually specified in ModelDesk after migration of the ModelDesk project. In the MATLAB/Simulink model, this is automatically done in the pre- and postmigrate infiles written by the ASM migration. After executing `go` for the second time in MATLAB (migration is completed), check the model structure for these values and enter them in the corresponding engine setup diesel page in ModelDesk. Model structure:

`MDL.EngineDiesel.Setup.Const_max_num_Cyl`

`MDL.EngineDiesel.Setup.Const_max_num_Inj_Port`

`MDL.EngineDiesel.Setup.Const_max_num_Inj_Direct`

# ASM Diesel Exhaust Blockset

## ASM Diesel Exhaust Blockset 2.0

---

**DIESEL\_OXIDATION\_  
CATALYST**

You can avoid negative lambda values by setting the value to 99 for the negative backflow through the DOC.



# ASM Diesel InCylinder Blockset

## Changes in the ASM Diesel InCylinder Demo Model

---

<b>/MDLUserInterface/ Environment/MDL_PAR</b>	The signal routing has been restructured.
<b>/MDL/Environment</b>	The signal routing has been restructured.
<b>/MDL/Environment/Driver</b>	The SignalSelection block is no longer linked to an ASM library. It is now a Simulink subsystem.

---

# ASM Drivetrain Basic Blockset

## Where to go from here

Information in this section

<i>New Features of ASM Drivetrain Basic Blockset 4.0</i>	50
<i>Migrating to ASM Drivetrain Basic Blockset 4.0</i>	50

## New Features of ASM Drivetrain Basic Blockset 4.0

### TORQUE\_CONTROLLER

The TORQUE\_CONTROLLER block can now also be activated if a negative torque set is provided.

### SIGNAL\_SELECTION

The SIGNAL\_SELECTION block has been removed from the library and is kept in the demos as a regular Simulink subsystem.

The former version of the block is also kept in the Former Versions sublibrary:

ASM\_DrivetrainBasic\_lib/Driver/FormerVersions/SIGNAL\_SELECTION\_4\_0

### CYCLES

The integrator inside the CYCLES block is changed from continuous to discrete to ensure a unified step size for the provided signals.

### LUT1D

The LUT1D block is new. It can be used to include measurement data in a model. For the block, a new subsystem has been created in ASM\_DrivetrainBasic\_lib/Driver/Measurement.

## Migrating to ASM Drivetrain Basic Blockset 4.0

### TORQUE\_CONTROLLER

During the migration, a set of blocks is added to the Sw\_TrqController\_Mode[0Off|1On] inport, so that the block is inactive if there is a negative torque set. This ensures the old functionality of the block.

---

**SIGNAL\_SELECTION**

During the migration, the link to the block is changed to the Former Versions sublibrary. It can now be found in  
ASM\_DrivetrainBasic\_lib/Driver/FormerVersions/  
SIGNAL\_SELECTION\_4\_0.

# ASM Electric Components Blockset

## Where to go from here

Information in this section

<i>New Features of ASM Electric Components Blockset 2.7</i>	52
<i>Changes in the ASM Electric Components Demo Model</i>	52

## New Features of ASM Electric Components Blockset 2.7

### **ELECTRIC\_MACHINE\_BASIC**

The **ELECTRIC\_MACHINE\_BASIC** block is new. It represents the basic function of an electric machine (e.g., PMSM or SCIM) with a current controller and power electronics. The electric machine is simulated as a first-order transfer function (PT1).

### **THREE\_PHASE\_DCM\_INVERTER**

The **THREE\_PHASE\_DCM\_INVERTER** block is new. It represents a three-phase discontinuous mode power converter that consists of up to six power switches connected in a bridge configuration.

### **SQUIRREL\_CAGE\_ASYNCHRONOUS\_MACHINE\_D\_Q**

The Back EMF Voltage output is now calculated. New signals (e.g., q-axis back EMF voltage) have been added to the **ASMSignalBus**.

### **SCIM\_CONTROLLER**

A fieldweakening controller has been added. The squirrel cage induction machine can now operate above the base speed range.

## Changes in the ASM Electric Components Demo Model

### **Vehicle Electrical System Demo**

The **ELECTRIC\_MACHINE\_BASIC** block has been added to the **ASM\_VehicleElectricalSystem** demo model as additional load.

### **SQUIRREL\_CAGE\_ASYNCHRONOUS\_MACHINE\_D\_Q**

This is a new demo model of the squirrel cage induction machine.

# ASM Engine Gasoline Basic Blockset

## Where to go from here

Information in this section

<i>New Features of ASM Engine Gasoline Basic Blockset 2.0</i>	53
<i>Changes in the ASM Engine Gasoline Basic Demo Model</i>	54
<i>Migrating to ASM Engine Gasoline Basic Blockset 2.0</i>	54

## New Features of ASM Engine Gasoline Basic Blockset 2.0

### THROTTLE\_MECHANICAL

The THROTTLE\_MECHANICAL block is new for the calculation of the throttle valve position according to a control signal.

A switch for Pos\_Throttle[%] has been added to the MDL\_In block:

- When switching on `SoftECU`, the throttle position is calculated by the THROTTLE\_MECHANICAL block.
- When switching on `RealECU`, the throttle position must be defined from outside the model, for example, by a real throttle valve or an external model.

### THROTTLE\_VALVE

The THROTTLE\_VALVE block is the successor of the THROTTLE block. Among other modifications, the `Map_A_Red` parameter of the THROTTLE\_VALVE block has been scaled to one.

### ENGINE\_SETUP

The ENGINE\_SETUP block is new. It contains the basic mechanical parameters of the engine. It also contains parameters used to control the engine and set the dimensions of the signals used inside the EngineGasoline Basic model.

### COMMON\_ENGINE\_PARAMETERS

The COMMON\_ENGINE\_PARAMETERS block is new. It contains the physical constants of air, exhaust, and fuel.

The COMMON\_GASOLINE\_PARAMETERS block is discontinued. It is split up into the COMMON\_ENGINE\_PARAMETERS and the ENGINE\_SETUP blocks.

---

<b>FRICION_TORQUE</b>	The look-up table in the <b>FRICION_TORQUE</b> block is now calculated based on friction torque and not based on friction pressure.
-----------------------	---

## Changes in the ASM Engine Gasoline Basic Demo Model

---

<b>/MDLUserInterface/ Environment/MDL_PAR</b>	The signal routing has been restructured.
<b>/MDL/Environment</b>	The signal routing has been restructured.
<b>/MDL/Environment/Driver</b>	The <b>SignalSelection</b> block is no longer linked to an ASM library. It is now a Simulink subsystem.
<b>Cooling System</b>	The signal routing has been restructured.
<b>Throttle valve position</b>	A switch for <b>Pos_Throttle[%]</b> has been added to the <b>MDL_In</b> block: <ul style="list-style-type: none"><li>■ When switching on <b>SoftECU</b>, the throttle position is calculated by the <b>THROTTLE_MECHANICAL</b> block.</li><li>■ When switching on <b>RealECU</b>, the throttle position must be defined from outside the model, for example by a real throttle valve or an external model.</li></ul>

## Migrating to ASM Engine Gasoline Basic Blockset 2.0

---

<b>COMMON_GASOLINE_ PARAMETERS_8_0</b>	<p>The <b>COMMON_GASOLINE_PARAMETERS</b> block no longer exists. It is replaced by two new blocks: <b>COMMON_ENGINE_PARAMETERS</b> and <b>ENGINE_SETUP</b>. The original parameters are assigned to the new blocks.</p> <p>The <b>COMMON_GASOLINE_PARAMETERS</b> block has been converted to a regular Simulink subsystem <b>Common_Gasoline_Parameters</b> containing the new blocks.</p>
--	--

**THROTTLE\_2\_0**

The THROTTLE block no longer exists. It is replaced by two new blocks: THROTTLE\_MECHANICAL and THROTTLE\_VALVE. The original parameters are assigned to the new blocks.

The THROTTLE block has been converted to a regular Simulink subsystem containing the new blocks.

**COMMON\_ENGINE\_PARAMETERS**

The following parameters are added in the premigrate variant:

- Const\_Rm
- Const\_Q\_LHV
- Const\_BulkModulus\_Fuel
- Const\_Fuel\_Coeff
- Const\_Air\_Coeff
- Const\_Exhaust\_Coeff
- Const\_R\_Fuel
- Const\_R\_Exhaust

**ENGINE\_SETUP**

The following parameters are added in the premigrate variant:

- Const\_max\_num\_Inj\_Direct
- Const\_max\_num\_Inj\_Port
- Const\_Inj\_Matrix
- Const\_Ratio\_Comp
- Const\_A\_Piston
- Const\_FiringOrder
- Const\_num\_Inj\_Direct

The MDL.EngineGasolineBasic.Const.Const\_max\_num\_Cyl parameter value is assigned to the MDL.EngineGasolineBasic.Setup.Const\_max\_num\_Cyl parameter during postmigration. The MDL.EngineGasolineBasic.Const.Const\_max\_num\_Inj.v parameter value is assigned to MDL.EngineGasolineBasic.Setup.Const\_max\_num\_Inj\_Direct during postmigration.

The Const\_max\_num\_Inj\_Direct, Const\_max\_num\_Inj\_Port and Const\_max\_num\_Cyl parameters are used to match the signal sizes in the engine model to the external model or the real ECU in the MDL\_In system. Also, the ASM models from the Operator libraries will use these values to define the dimensional input to the S-function placed under their masks.

The functionality to write these values with ModelDesk is new. Therefore they have to be manually specified in ModelDesk after migration of the ModelDesk project. In the MATLAB/Simulink model, this is automatically done in the pre- and postmigrate inifiles written by the ASM migration. After executing `go` for the second time in MATLAB (migration is completed), check the model structure for these values and enter them in the corresponding engine setup gasoline page in ModelDesk. Model structure:

`MDL.EngineDiesel.Setup.Const_max_num_Cyl`

`MDL.EngineDiesel.Setup.Const_max_num_Inj_Port`

`MDL.EngineDiesel.Setup.Const_max_num_Inj_Direct`

---

### THROTTLE\_VALVE

The `Const_A_max` parameter is added in the premigrate variant.



# ASM Engine Gasoline Blockset

## Where to go from here

Information in this section

<i>New Features of ASM Engine Gasoline Blockset 3.0</i>	57
<i>Changes in the ASM Engine Gasoline Demo Model</i>	58
<i>Migrating to ASM Engine Gasoline Blockset 3.0</i>	59

## New Features of ASM Engine Gasoline Blockset 3.0

### HPP\_CRANKBASED

The HPP\_CRANKBASED block includes a new model for a high-pressure pump of a fuel system. It calculates the volume flow through the high-pressure pump as a function of the crank angle.

The block contains two model approaches:

- A mean value model
- A pulsed model

Both approaches use the fuel delivery time as the source of volume calculation. The control signal of this pump is the start of the energization (closing) of the fuel metering unit in degrees of the crank angle.

### RAIL

The block has a new parameter: `Const_T_Rail`.

### PUMP\_TORQUE

The calculation of the compression and the implementation of the piston pressure difference have been modified.

### THROTTLE\_MECHANICAL

This is a new block for the calculation of the throttle valve position according to a control signal.

A switch for `Pos_Throttle[%]` has been added to the `MDL_In` block:

- When switching on `SoftECU`, the throttle position is calculated by the `THROTTLE_MECHANICAL` block.
- When switching on `RealECU`, the throttle position must be defined from outside the model, for example, by a real throttle valve or an external model.

---

<b>THROTTLE_VALVE</b>	This block is the successor of the THROTTLE block. Among other modifications, the <code>Map_A_Red</code> parameter of the THROTTLE_VALVE block has been scaled to one.
<b>ENGINE_SETUP</b>	The ENGINE_SETUP block is new. It contains the basic mechanical parameters of the engine. It also contains parameters used to control the engine and set the dimensions of the signals used inside the EngineGasoline model.
<b>COMMON_ENGINE_PARAMETERS</b>	This block is new. It contains the physical constants of air, exhaust, and fuel. The <code>COMMON_GASOLINE_PARAMETERS</code> block is discontinued. It is split up into the <code>COMMON_ENGINE_PARAMETERS</code> and <code>ENGINE_SETUP</code> blocks.
<b>RAIL_CONTROL_CRANKBASED</b>	<p>This block is new. It serves as the controller for the <code>HPP_CRANKBASED</code> block.</p> <p>It controls the rail pressure by calculating the fuel metering unit's (FMU's) set point of actuation. The rail pressure set point depends on the engine's operation point. A constant or an external set point can be used for testing.</p> <p>The controller is a PI controller with subsequent linear control to angle conversion. A trigger mode can be enabled to calculate the start of the FMU actuation before the working cycle and to keep that starting point until the next pump's working cycle.</p>
<b>FRICITION_TORQUE</b>	The look-up table in the <code>FRICITION_TORQUE</code> block is calculated as a map for friction torque and not as a map for friction pressure.

---

## Changes in the ASM Engine Gasoline Demo Model

---

<b>FuelSystem</b>	<p>The <code>FuelSystem</code> model now contains two high-pressure pump models, one current-based and one crank-angle-based.</p> <p>You can open the <code>High-Pressure Pump</code> library by clicking <code>Open High Pressure Pump Models</code> next to the <code>HighPressurePump</code> subsystem in the model. You can replace the <code>HighPressurePump</code> model via drag &amp; drop of the block from the library.</p>
<b>/MDLUserInterface/Environment/MDL_PAR</b>	The signal routing has been restructured.

---

<b>/MDL/Environment</b>	The signal routing has been restructured.
<b>/MDL/Environment/Driver</b>	The SignalSelection block is no longer linked to an ASM library. It is now a Simulink subsystem.
<b>Cooling System</b>	The signal routing has been restructured.
<b>Throttle valve position</b>	<p>A switch for Pos_Throttle[%] has been added to the MDL_In block:</p> <ul style="list-style-type: none"> <li>■ When switching on SoftECU, the throttle position is calculated by the THROTTLE_MECHANICAL block.</li> <li>■ When switching on RealECU, the throttle position must be defined from outside the model, for example, by a real throttle valve or an external model.</li> </ul>
<b>HighPressurePump</b>	The default high-pressure pump model is now HPP_CRANKBASED (crank-angle-based high-pressure pump) and not HIGH_PRESSURE_PUMP (current-based high-pressure pump).

## Migrating to ASM Engine Gasoline Blockset 3.0

<b>COMMON_GASOLINE_PARAMETERS_8_0</b>	<p>The COMMON_GASOLINE_PARAMETERS block no longer exists. It is replaced by two new blocks: COMMON_ENGINE_PARAMETERS and ENGINE_SETUP. The original parameters are assigned to the new blocks.</p> <p>The COMMON_GASOLINE_PARAMETERS block has been converted to a regular Common_Gasoline_Parameters Simulink subsystem containing the new blocks.</p>
<b>THROTTLE_4_0</b>	<p>The THROTTLE block no longer exists. It is replaced by two new blocks: THROTTLE_MECHANICAL and THROTTLE_VALVE. The original parameters are assigned to the new blocks.</p> <p>The THROTTLE block has been converted to a regular Simulink subsystem containing the new blocks.</p>
<b>SWITCHES_FUEL_METER_UNIT_1_0</b>	The block has been discontinued. The block's parameter has been moved to the HIGH_PRESSURE_PUMP block.
<b>SWITCHES_THROTTLE_1_0</b>	The block has been discontinued.

---

### HIGH\_PRESSURE\_PUMP

The parameter of the discontinued SWITCHES\_FUEL\_METER\_UNIT\_1\_0 block has been moved to the HIGH\_PRESSURE\_PUMP block. In former versions, the SWITCHES\_FUEL\_METER\_UNIT parameter was an inport to the HIGH\_PRESSURE\_PUMP model. Thus, in the postmigrate variant of the HIGH\_PRESSURE\_PUMP, the setting of the new to old parameter structure takes place and the old inport is terminated in the model. The Map\_V\_dead and Const\_p\_Low parameters are added during premigration.

---

### ENGINE\_SETUP

The Const\_max\_num\_Inj\_Direct and Const\_max\_num\_Inj\_Port are added in the premigrate variant.

The parameter value of MDL.EngineGasoline.Const.Const\_max\_num\_Cyl is assigned to the MDL.EngineGasoline.Setup.Const\_max\_num\_Cyl parameter in the postmigrate variant.

The parameter value of MDL.EngineGasoline.Const.Const\_max\_num\_Inj.v is assigned to MDL.EngineGasoline.Setup.Const\_max\_num\_Inj\_Direct in the postmigrate variant.

The Const\_max\_num\_Inj\_Direct, Const\_max\_num\_Inj\_Port and Const\_max\_num\_Cyl parameters are used to match the signal sizes in the engine model to the external model or the real ECU in the MDL\_In system. Also, the ASM models from the Operator libraries will use these values to define the dimensional input to the S-function placed under their masks.

The functionality to write these values with ModelDesk is new. Therefore they have to be manually specified in ModelDesk after migration of the ModelDesk project. In the MATLAB/Simulink model, this is automatically done in the pre- and postmigrate inifiles written by the ASM migration. After executing go for the second time in MATLAB (migration is completed), check the model structure for these values and enter them in the corresponding engine setup gasoline page in ModelDesk. Model structure:

MDL.EngineDiesel.Setup.Const\_max\_num\_Cyl

MDL.EngineDiesel.Setup.Const\_max\_num\_Inj\_Port

MDL.EngineDiesel.Setup.Const\_max\_num\_Inj\_Direct

---

### RAIL

The Const\_T\_Rail parameter is added in the premigrate variant.

---

**COMMON\_ENGINE\_PARAMETERS**

The Const\_Rm and Const\_Q\_LHV parameters are added in the premigrate variant.

---

**THROTTLE\_VALVE**

The Const\_A\_max is added in the premigrate variant.

---

**Related topics**

Basics

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

# ASM Gasoline InCylinder Blockset

## Where to go from here

Information in this section

<i>Changes in the ASM Gasoline InCylinder Demo Model</i>	62
<i>Migrating to ASM Gasoline InCylinder Blockset 1.9</i>	62

## Changes in the ASM Gasoline InCylinder Demo Model

### /MDLUserInterface/ Environment/MDL\_PAR

The signal routing has been restructured.

### /MDL/Environment

The signal routing has been restructured.

### /MDL/Environment/Driver

The SignalSelection block is no longer linked to an ASM library. It is now a Simulink subsystem.

## Migrating to ASM Gasoline InCylinder Blockset 1.9

### HEAT\_RELEASE\_VIBE

The Const\_num\_Cyl mask parameter has been corrected to Const\_max\_num\_Cyl.

The initialization of several memory blocks has been changed to use the mask parameter.

ones(MDL.InCylinderGasoline.Setup.Const\_max\_num\_Cyl.v,1) has been replaced by ones(Const\_max\_num\_Cyl.v,1).

### CATALYST

The PT2 transfer function has been updated. This affects the realtime path of the T1 parameter.

# ASM Parameterization Tool

## Where to go from here

Information in this section

<i>New Features of the ASM Parameterization Tool 1.6.5</i>	63
<i>Migrating to ASM Parameterization Tool 1.6.5</i>	63

## New Features of the ASM Parameterization Tool 1.6.5

### ModelDesk Processing

ModelDesk now provides the *Processing* functionality. It supports the execution of calculation functions together with the administration of measurement information.

With this new feature, ModelDesk supports the ASMPParameterization Tool functionalities, among others. As of this release, new users of mean value engine models are required to start an engine parameterization project in ModelDesk, because no demo engine parameterizations are provided for mean value engine models. Nevertheless, the ASMPParameterization Tool is still provided in the dSPACE installation, allowing the migration from previous releases. In this way, users are able to use existing parameterization projects also in the current release.

## Migrating to ASM Parameterization Tool 1.6.5

### ModelDesk export

The Parameter Export to ModelDesk page is no longer displayed. The user is requested to use an import functionality provided with the new ModelDesk Processing feature.

Importing parameters in ModelDesk is supported by the GeneralSettings file with the following content:

```
****
```

```
function Settings = MySettings()
```

```
Settings.PreFcn = '<Filepath with filename and extension of  
ImportMDLfromWorkspace.m>';
```

\*\*\*

Make sure to copy the `ImportMDLfromWorkspace.m` file from the installed engine parameterization demo zip file beforehand:

```
<dSPACE_Root>\Demos\ASM\<EnginePackage>\Parameterization\  
<ModelDesk-Project.zip>\Pool\Processing\Function\  
PreFcns\ImportMDLfromWorkspace.m.
```

When you evaluate the processing configuration in ModelDesk, the command reads the current MATLAB workspace structure and imports the existing parameter to the ModelDesk parameter set of the simulation model used in the ModelDesk experiment. Evaluation of the processing configuration is done via the corresponding button on the Processing Configuration page in ModelDesk.



# ASM Pneumatics Blockset

## Where to go from here

Information in this section

<i>New Features of ASM Pneumatics Blockset 2.0</i>	65
<i>Migrating to ASM Pneumatics Blockset 2.0</i>	65

## New Features of ASM Pneumatics Blockset 2.0

### Air Suspension

The new Air Suspension sublibrary contains blocks for the implementation of air suspension systems for trucks or trailers:

- Air suspension valves
- Pipes
- Air springs and lifting bellows for front and rear axles

Furthermore, there are Setup blocks allowing a wide range of different air suspension configurations.

### BrakePneumatics\_ AirSuspension

This demo model is new. It consists of air supply, brake pneumatics and air suspension for a vehicle with three axles.

### BrakePneumatics\_ AirSuspension\_Trailer

This demo model is new. It consists of air supply, brake pneumatics and air suspension for a trailer with three axles.

## Migrating to ASM Pneumatics Blockset 2.0

### COMMON\_PARAMETERS

The block has the new `Const_cv_Air` and `Const_kappa_Air` parameters for the isochore heat capacity and the isentropic exponent of air.

The block has new outputs for the isochore heat capacity and the isentropic exponent of air.

# ASM Traffic Blockset

## Where to go from here

Information in this section

<i>New Features of ASM Traffic Blockset 3.1</i>	66
<i>Migrating to ASM Traffic Blockset 3.1</i>	66

## New Features of ASM Traffic Blockset 3.1

### SOFT\_ECU\_ACC

The velocity controller has been replaced by a new PI controller with precontrol functionality. The minimal and maximal value and rate of the acceleration and deceleration are now set dynamically and depend on the vehicle velocity. Also, an AEB (Autonomous Emergency Brake) and FCW (Forward Collision Warning) system are added.

## Migrating to ASM Traffic Blockset 3.1

### SOFT\_ECU blocks

The velocity controller has been replaced with a controller with new functionality. An AEB (Autonomous Emergency Brake) and an FCW (Forward Collision Warning) system are also added, so that parameters cannot be migrated automatically.

The link to the SOFT\_ECU sublibrary is changed to the former implementation (see FormerVersion subsystem) during the migration of older ASM models. Thus, the simulation behavior is not changed. If you want to use the new SOFT\_ECU implementation, drag the blocks from the ASM\_Traffic\_lib library to your model and adapt the parameters to your needs.

This applies to the following blocks:

- SOFT\_ECU\_ACC
- TARGET\_SELECTION\_ACC
- USER\_INTERFACE\_ACC
- OUTPUT\_INTERFACE\_ACC

# ASM Turbocharger Blockset

## New Features of ASM Turbocharger Blockset 3.0

---

### **POSTTURBHPMAN**

The specific enthalpy is now available as an output of the block.

# ASM Vehicle Dynamics Blockset

## Where to go from here

Information in this section

<i>New Features of ASM Vehicle Dynamics Blockset 3.0</i>	68
<i>Changes in the ASM Vehicle Dynamics Demo Model</i>	70
<i>Migrating to ASM Vehicle Dynamics Blockset 3.0</i>	70

## New Features of ASM Vehicle Dynamics Blockset 3.0

### SUSCOMP\_2D\_FRONT

This new suspension compliance block can be used to parameterize a compliance which also depends on the force and torque at the left wheel and the right wheel.

### SUSCOMP\_2D\_REAR

This new suspension compliance block can be used to parameterize a compliance which also depends on the force and torque at the left wheel and the right wheel.

### TIRE\_VEL\_SAMPLING\_POINTS

The Magic Formula tire model can be used to parameterize different tire slip curves depending on the longitudinal tire velocity. Four velocity sampling points can be defined in this block.

### STEERING\_3DOF\_VARIABLE\_RATIO

The STEERING\_3DOF\_VARIABLE\_RATIO block is an improvement of the existing steering block with three inertial masses, corresponding force elements, and new friction elements. Compared to the 1DOF model, the on-center handling behavior of a vehicle is better represented by the 3DOF model. Also, the block can be parameterized with variable steering ratios.

### SOFT\_ECU\_POWERSTEERING

The SOFT\_ECU\_POWERSTEERING block has been added to the Soft ECU blockset to provide the assistive power steering torque to the steering model.

## Drivetrain

The Drivetrain model has been expanded with a new modeling approach. Besides the original flexible drivetrain, i.e., with elastic shafts, a rigid drivetrain can now be simulated. The main differences between the two approaches are described in the following:

**Flexible Drivetrain:** Simulates the behavior of a drivetrain with stiff drive shafts and takes the elasticity and the inertias of the drivetrain components into account.

By including the stiff drive shafts, the drivetrain dynamics and oscillations can be simulated. However, the relatively high stiffness and small inertias of the different components can result in a stiff system of equations. This raises the difficulty of the numerical integration and might lead to an unstable system. Therefore, in this case, the complete drivetrain model is stabilized.

**Rigid Drivetrain:** Simulates the behavior of a drivetrain without stiff drive shafts. No elasticity or inertias of the drivetrain components are taken into account.

Using this model approach, you can simulate the drivetrain components with simpler models and fewer parameters. It is also possible to extend the standard drivetrain model for other configurations with relatively low effort.

For the rigid drivetrain, several new blocks have been introduced:

- CENTRAL\_DIFFERENTIAL\_RIGID
- REAR\_DIFFERENTIAL\_RIGID
- FRONT\_DIFFERENTIAL\_RIGID
- CLUTCH\_RIGID
- CLUTCH\_4WD\_RIGID
- TORQUE\_CONVERTER\_RIGID
- GEARBOX\_MT\_RIGID
- GEARBOX\_AT\_RIGID
- TRANSFER\_GEARBOX\_RIGID
- CRANK\_SHAFT\_RIGID
- LOCKUP\_CLUTCH\_RIGID

Different final drive assembly configurations are supported and offered as standard demos for both flexible and rigid drivetrain. You can switch between these configurations by exchanging specific subsystems and adapting the related settings and switches.

## Changes in the ASM Vehicle Dynamics Demo Model

---

### Suspension Compliance

The new `SUSCOMP_2D_FRONT` block is used as default suspension compliance for the front axle.

The new `SUSCOMP_2D_REAR` block is used as default suspension compliance for the rear axle.

---

### Steering

The new `STEERING_3DOF_VARIABLE_RATIO` model is used as the default steering model.

The `SOFT_ECU_POWERSTEERING` block has been added to the SoftECU subsystem.

The `MDL_IN` block has been extended to control the assistive torque provided to the steering model.

---

### Drivetrain subsystem

Due to the new rigid drivetrain approach, new shortcuts have been added inside the Drivetrain subsystem.

These shortcuts open the following drivetrain-related demos for both flexible and rigid drivetrain:

- Crankshaft demos
- Transmission System demos
- Final Drive Assembly demos

## Migrating to ASM Vehicle Dynamics Blockset 3.0

---

### `TIRE_MODEL_MAGIC_` `FORMULA_xxx`

Two new inports for tire parameters are added which depend on the longitudinal velocity.

The `ScalingParameter.LFZO` parameter in the XML file (for ModelDesk) has been renamed `ScalingParameters.LFZO`.

---

### `CRANK_SHAFT`

The `CRANK_SHAFT` block has the new `Sw_Trq_Mass_Mode` parameter. This parameter was provided via signal routing and inport. The deletion of the corresponding inport is compensated during migration.

---

**DRIVETRAIN\_VARIANT\_  
SWITCHES**

The Sw\_Trq\_Mass\_Mode parameter has been deleted. This parameter was provided to the CRANK\_SHAFT block via signal routing and inport. The CRANK\_SHAFT block now includes this parameter instead of getting it as input.

---

**AERODYNAMICS**

The calculation of the wind force now takes the vertical wind velocity ( $v_z$ ) into account. Thus, the amount of the wind vector during pitch is considered correctly.





---

# ConfigurationDesk

---

**Objective**

ConfigurationDesk is a tool that can be applied in different use scenarios. You can use it to implement real-time applications and configure RapidPro hardware.

# ConfigurationDesk – Implementation

---

**Where to go from here**

Information in this section

<i>New Features of ConfigurationDesk 5.2 (Implementation Version)</i>	74
<i>Migrating to ConfigurationDesk 5.2</i>	79

---

## New Features of ConfigurationDesk 5.2 (Implementation Version)

---

**New methods for modeling asynchronous tasks**

ConfigurationDesk provides new methods to connect I/O events with tasks and runnable functions more conveniently:

- If your ConfigurationDesk application already contains an application process with a task, you can easily assign an I/O event to that task in the Properties Browser of the function block.
- If your ConfigurationDesk application contains an application process for a model without a suitable Runnable Function block, you can use the Generate Model Interface - Runnable Function Blocks and Tasks command that is available in the context menu of the function block. ConfigurationDesk then automatically creates not only a suitable Runnable Function block within a Simulink interface model, but also a task that contains the corresponding runnable function and the I/O event.

For details, refer to *Basics on Modeling Asynchronous Tasks* ( *ConfigurationDesk Real-Time Implementation Guide*).

**Creating preconfigured application processes directly when adding behavior models**

When you add a behavior model to your executable application, you can have ConfigurationDesk create a preconfigured application process directly for that behavior model. To do so, you have to select the Create preconfigured application process checkbox in the Add Model or the Add/Replace Model Topology dialog. If the executable application does not contain a processing unit application, the preconfigured application process is created in a new processing unit application. If the executable application contains exactly one processing unit application, the preconfigured application process is created in the existing processing unit application. The Create preconfigured application process checkbox is selected by default. For details, refer to *Add Model* ([📖 ConfigurationDesk Real-Time Implementation Reference](#)).

**Using application processes without a behavior model**

There are specific use cases in which you do not need to map the I/O functions to the ports of a behavior model. Some examples are bringing simulators into service or for wire testing. For these purposes, ConfigurationDesk lets you create specific application processes that provide a default task and that no model implementation has to be assigned to. For details, refer to *Introduction to Application Processes Without Behavior Models* ([📖 ConfigurationDesk Real-Time Implementation Guide](#)).

**Support of FMUs without source files**

ConfigurationDesk provides methods for you to convert FMUs with source files into FMUs without source files but with a SCALEXIO-compatible library file. You can use the converted FMUs in your executable application. For details, refer to *Creating Precompiled FMUs* ([📖 ConfigurationDesk Real-Time Implementation Guide](#)).


**New features of V-ECU implementations**

**Supported V-ECU Implementation container versions** The following table shows the tool versions that export V-ECU implementation containers, and the related container versions.

V-ECU implementation Container Exported From	Container Version
SystemDesk 3.x	1.0
SystemDesk 4.2	2.0
SystemDesk 4.3	2.1
TargetLink 3.5	1.0
TargetLink 4.0	2.1

For details on V-ECUs exported by SystemDesk, refer to *Configuring ECUs* on page 154.

### Support of V-ECU implementations containing LIN


**controllers** ConfigurationDesk lets you add V-ECU implementations containing one or more LIN controllers to your executable application. For details, refer to *Special Aspects of V-ECU Implementations Containing LIN Controllers* ( *ConfigurationDesk Real-Time Implementation Guide*).



The following limitations apply to ConfigurationDesk's LIN communication support:

- The LIN configuration can define only the LIN master and no LIN slaves.
- The LIN transport protocol is not supported.
- LIN node configuration services are not supported.


### Delaying the startup of a V-ECU

**implementation** ConfigurationDesk lets you build real-time applications where the start of the contained V-ECU implementations can be delayed. For details, refer to *Delaying the Start of V-ECU Implementations* ( *ConfigurationDesk Real-Time Implementation Guide*).

**Support of RTE interventions** The former limitation no longer applies. You can now use V-ECU Implementations that use RTE Interventions also with SCALEXIO.

---

#### Generating model interfaces into an existing Simulink model

ConfigurationDesk lets you generate the model interface for all unresolved or marked model port blocks directly into a Simulink model that you have added to your ConfigurationDesk application. For details, refer to *Handling the Model Interface* ( *ConfigurationDesk Real-Time Implementation Guide*).

---

#### Unsupported new features of MATLAB R2014a

The following new features introduced with MATLAB R2014a are not supported:

- Data dictionaries
  - Simulink is able to store design data that your model uses in a data dictionary as a permanent repository.

---

**Unsupported new features of MATLAB R2014b**

The following new features introduced with MATLAB R2014b are not supported:

- **Simulink Function subsystems**

You cannot use dSPACE blocks, for example, model port blocks from the dSPACE Model Port Block Library, in a Simulink Function subsystem. The contents of Simulink Function subsystem are not generated into the TRC file.

- **Model Templates**

You cannot use dSPACE blocks, for example, model port blocks from the dSPACE Model Port Block Library, in a model template.

---

**Enhanced function block types**

**SENT In function block** The SENT In function block provides the following new features according to the SAE J2716 JAN2010 SENT standard:

- Receiving serial messages (in short and enhanced serial message format) via a defined number of SENT messages.
- Using predefined data protocols for data reception to support specific sensor applications.

For basic information, refer to *SENT In* ([📖 ConfigurationDesk I/O Function Implementation Guide](#)).

**Lambda DCR, Lambda NCCR function blocks** Via the Lambda DCR and Lambda NCCR function blocks, you can now specify whether the sensor simulation works GND-based or electrically isolated from GND of the dSPACE hardware.

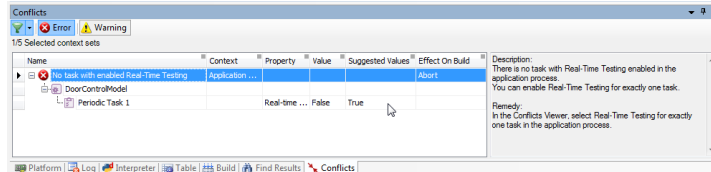
Using the Isolated configuration, you can support modern ECUs which require the sensor simulation to be similar to an ideal two-terminal circuit-generating with identical current flow on both signal terminals (signal and reference).

**All function blocks** The function ports of each function block now provide the *Model access* parameter. This parameter lets you enable/disable access to the behavior model via model port mapping.

You can set *Model access* to *Disabled* if you do not want to access the port value of a behavior model and you want to ensure that only required model ports are created for such function ports.

**Improved Conflicts Viewer**

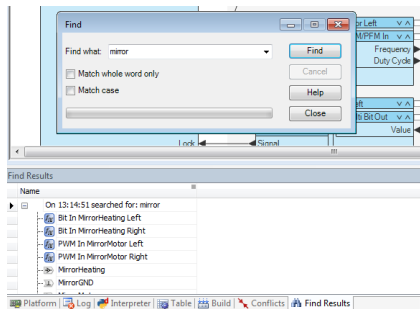
The Conflicts Viewer has been redesigned to make it easier for you to find conflict sources and to resolve conflicts. New filter functionalities let you focus on conflicts from specific context sets or on more or less severe conflicts.



For more information, refer to *Resolving Conflicts* ([ConfigurationDesk Real-Time Implementation Guide](#)).

**Global search**

The new Find command lets you search a ConfigurationDesk application for elements by name. You can search for any element whose properties can be accessed in the Properties Browser. Search results are displayed and organized in the Find Results Viewer.



For instructions, refer to *How to Find Elements of a ConfigurationDesk Application* ([ConfigurationDesk Real-Time Implementation Guide](#)).

**Collapsing and expanding blocks**

For a better handling of ports and mappings, you can now collapse and expand blocks in a graphical window using collapse and expand arrows.

For details, refer to *Collapsing and Expanding Blocks* ([ConfigurationDesk Real-Time Implementation Guide](#)).

**New method available for hardware resource assignment**

Now you can assign hardware resources (channel sets and/or channels) by dragging them from the available hardware topology to a specific function block. For details, refer to *Methods for Assigning Hardware Resources* ([ConfigurationDesk Real-Time Implementation Guide](#)).

## Migrating to ConfigurationDesk 5.2

---

### Pre-4.3 projects and component files no longer supported

ConfigurationDesk projects from ConfigurationDesk 4.2 or earlier cannot be migrated to ConfigurationDesk 5.2. Pre-4.3 Application component files (DTF, ECH, HTF, MTF) can also no longer be migrated.

---

### Modifications concerning the TRC file generation

The following settings in the Code Generation - DSRT variable description file options page of the Configuration Parameters dialog have been changed:

- The Apply subsystem read/write permissions setting is no longer available.
- The following settings are now cleared by default:
  - Include signal labels
  - Include virtual blocks

The new default values are only relevant for new models or if you switch your model to another platform.

- The Include only Simulink.Parameter and Simulink.Signal objects with global storage class setting now also considers the BusSystems group of the RTI CAN MultiMessage Blockset and the RTI LIN MultiMessage Blockset.

---

### Handling of Bus Selector block changed

With MATLAB R2014a, the Simulink Bus Selector block has been virtualized. In combination with dSPACE Release 2014-B, the outputs of this block and other virtual blocks that are connected to it are not generated into the variable description file if the inport of the Bus Selector block is connected to a non-virtual bus.

To access a signal of a bus do the following:

- Access the signal directly from the bus,  
or
- Connect a non-virtual block to the signal, for example, a Gain block with 1 as the factor. Its output is available in the variable description file.

If the inport of the Bus Selector block is connected via a virtual bus, the outputs are generated into the variable description file without applying special methods.

**Generated code from Simulink Coder has changed**

**Using MATLAB R2014a** With the MATLAB/Simulink R2014a release, code generation by Simulink Coder has changed. Therefore, also the generation of variable descriptions (TRC files) in ConfigurationDesk has changed.

Using dSPACE Release 2014-B in combination with MATLAB R2014a requires to use the `revertInlineParametersOffToR2013b` command before you start a build process. This command enables the Simulink Coder behavior and dSPACE TRC file generation in MATLAB R2014a similar to that in MATLAB R2013b and before. The `revertInlineParametersOffToR2013b` command is shipped with MATLAB R2014a as part of the Simulink Coder product.

For more details, see the Simulink Coder release notes: <http://www.mathworks.de/de/help/rtw/release-notes.html> (Mathworks Account required).

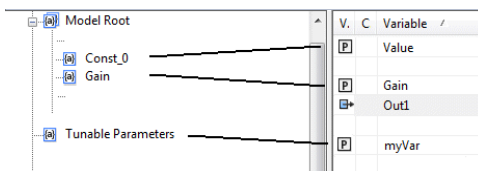
**Using MATLAB R2014b** The modification in the Simulink Coder affects the entries in the generated TRC file only if you have worked with the Inline Parameters optimization option cleared. Workspace variables that are referenced by one or more block parameters are now handled as global parameters in the model and not as local parameters of the block. They are stored to the Tunable Parameters group in the generated TRC file. If you now change the value of such a global parameter, any block parameter referencing this global parameter changes as well. The behavior of your simulation will change.

Example:

Workspace variables are generated as local block parameters in the following cases:

- The parameter is a structure or a structure item, for example, `myStruct.Value`.
- The parameter is used in an expression or in a function, for example, `myValue + 1` or `sin(myValue)`.

Local block parameters are also generated if references to mask parameters are used. The mask parameter itself is not generated into the variable description.





In the model, you have configured the `Const_0` block parameter and the `Gain` block parameter by using the workspace variable `myVar`. Independently of the `Inline parameters` option on the `Optimization` page, the workspace variable is generated into the variable description. The variables can be used in ControlDesk Next Generation as before by connecting `Const_0/Value` and `Gain/Gain` to instruments, but internally, the parameter values depend on the value of `myVar`. If you modify one of the parameters, all the other parameters are changed too.

### Migrating a custom function block

If some or all custom function block types are unresolved after loading a project, you can perform one of the following actions:

- On the `Configuration` page of the `ConfigurationDesk Options` dialog, change the global search path for custom function blocks to a folder containing the missing custom function XML files. The associated custom function block types are resolved automatically.

or

- Copy the XML file of each custom function block type either to the project-specific search path or to the global search path. If available, also copy the following files:
  - The header file: `<Function_block_type_name/CModule_name>.h`
  - The C++ source code file: `<Function_block_type_name/CModule_name>.cpp`
  - The type definition file: `<Function_block_type_name/CModule_name>_TypeDef.h`

The project-specific custom function directory is:

```
<DocumentsFolder>\<Project>\CustomFunctions
```

The default global search path for custom functions is:

```
<DocumentsFolder>\UserFiles
```

Afterwards, you can resolve the function block types via the `Reload Custom Function Definitions` command in the `Function Browser`.

If an existing header or source file is not found, open the corresponding XML file of the custom function block type with a suitable editor and make sure that in the `<CModule Name="XXX">` tag, `XXX` is the same as the `<Function_block_type_name/CModule_name>`.

If you only have the XML file, you can create the C++ source code file and the header file via `Create Custom Function Code`, and the type definition file via `Create Custom Function Type Definition` from the context menu of a custom function block type.

---

**Changed custom function path in variable description file**

The path to the variables of custom functions in the variable description file has been changed. For example, when you reload a variable description file from ConfigurationDesk 5.2 in ControlDesk Next Generation the connections of layouts and signal generators to these variables are lost. You must reconnect these variables.

---

# ControlDesk Next Generation



---

## Where to go from here

Information in this section

<i>New Features of ControlDesk Next Generation (ControlDesk 5.3)</i>	84
<i>Migrating to ControlDesk Next Generation (ControlDesk 5.3)</i>	93

Information in other sections

 <i>ControlDesk Next Generation Migration Guide</i> Explains migration from ControlDesk 3.x, CalDesk and prior versions of ControlDesk Next Generation to ControlDesk 5.3.
 <i>ControlDesk Next Generation Migration of ControlDesk 3.x Automation</i> Explains migration from ControlDesk 3.x automation to ControlDesk Next Generation automation.

# New Features of ControlDesk Next Generation (ControlDesk 5.3)

## Where to go from here

Information in this section

<i>New General Features (ControlDesk 5.3)</i>	84
<i>New Project and Experiment Features (ControlDesk 5.3)</i>	85
<i>New Features of Platform Management and Platforms/Devices (ControlDesk 5.3)</i>	85
<i>New Variable Management Features (ControlDesk 5.3)</i>	86
<i>New Visualization and Instrument Features (ControlDesk 5.3)</i>	86
<i>New Measurement and Recording Features (ControlDesk 5.3)</i>	89
<i>New Bus Navigator Features (ControlDesk 5.3)</i>	90
<i>New Data Set Management Features (ControlDesk 5.3)</i>	90
<i>New ECU Diagnostics Features (ControlDesk 5.3)</i>	91
<i>New Signal Editor Features (ControlDesk 5.3)</i>	91
<i>New Automation Features (ControlDesk 5.3)</i>	92

## New General Features (ControlDesk 5.3)

### Customizing the ribbon

You can customize ControlDesk's ribbon by adding custom ribbon controls. You have to link each custom ribbon control to a Python *extension script*, which allows you to add custom functions based on ControlDesk's automation interface to the user interface.

Refer to *Customizing the Ribbon* ( *ControlDesk Next Generation Basic Practices Guide*).

## New Project and Experiment Features (ControlDesk 5.3)

### Support of different SYNECT server versions

When you use ControlDesk in connection with the SYNECT server, ControlDesk now lets you work with SYNECT server versions of different dSPACE Releases (starting with the SYNECT server version in dSPACE Release 2014-B).

## New Features of Platform Management and Platforms/Devices (ControlDesk 5.3)

### MicroLabBox support

ControlDesk supports the new MicroLabBox. See also *New Features of RTI/RTI-MP and RTLib* on page 119.

### Support of Kvaser Leaf Light HS interface for CAN

ControlDesk now supports the Leaf Light HS interface from Kvaser for CAN.

Refer to *Supported CAN Interfaces* ([📖 ControlDesk Next Generation Basic Practices Guide](#)).

### Platform seeking when starting ControlDesk via automation

You can now specify the platform seeking options also for starting ControlDesk *via automation*. Platform seeking means searching for registered platforms when ControlDesk starts.

For details, refer to *Platform Management Page* ([📖 ControlDesk Next Generation Reference](#)).

### Accessing dSPACE real-time hardware with corrupted boot firmware

As of ControlDesk 5.3, dSPACE real-time hardware with corrupted boot firmware is displayed in ControlDesk's Platform/Device Manager and indicated with the 🚩 symbol.

You should check the Log Viewer entries and perform a firmware update if necessary. Refer to *Update Firmware* ([📖 ControlDesk Next Generation Reference](#)).

### Enhancements to memory segment management

- ControlDesk now supports all the memory segment types in an A2L file.
- You can now specify whether the content of each individual memory segment is to be compared with the segment content of the device's mirrored memory when online calibration is started.
- You can now specify *individually for each memory segment* whether it is to be exported when an ECU Image file is generated.

For details, refer to *Memory Segments* ([📖 ControlDesk Next Generation Reference](#)).

**Video Capturing device:  
Playing data from video  
streams**

You can now simulate a real camera by playing video streams synchronously to signals from other platforms during measurement. Refer to *How to Configure Video Capturing Devices* ([📖 ControlDesk Next Generation Basic Practices Guide](#)).

**VEOS platform: New  
automation event when  
pausing simulation**

The VEOS platform now provides an event when a running offline simulation is paused.

**DS1005-based and  
DS1006-based  
multiprocessor systems:  
Editing processor names**

You can now specify processor names for the boards belonging to a DS1005-based or DS1006-based multiprocessor system. Refer to *Edit Processor Names* ([📖 ControlDesk Next Generation Reference](#)).

## New Variable Management Features (ControlDesk 5.3)

**Avoiding long project  
loading times**

To reduce the loading time of a project, you can compress the content of the database containing the project's variable descriptions. This removes unnecessary data from the database.

Refer to *Problem with Long Project Loading Times* ([📖 ControlDesk Next Generation Basic Practices Guide](#)).

## New Visualization and Instrument Features (ControlDesk 5.3)

**Information in this topic**

<i>Time Plotter</i>	86
<i>Adding a Python script to an instrument</i>	87
<i>Stopwatch instruments</i>	88
<i>Automating the Instrument Selector</i>	88
<i>Improvements to the selection of instruments</i>	88
<i>Placing all the variables of a subgroup on a layout</i>	89

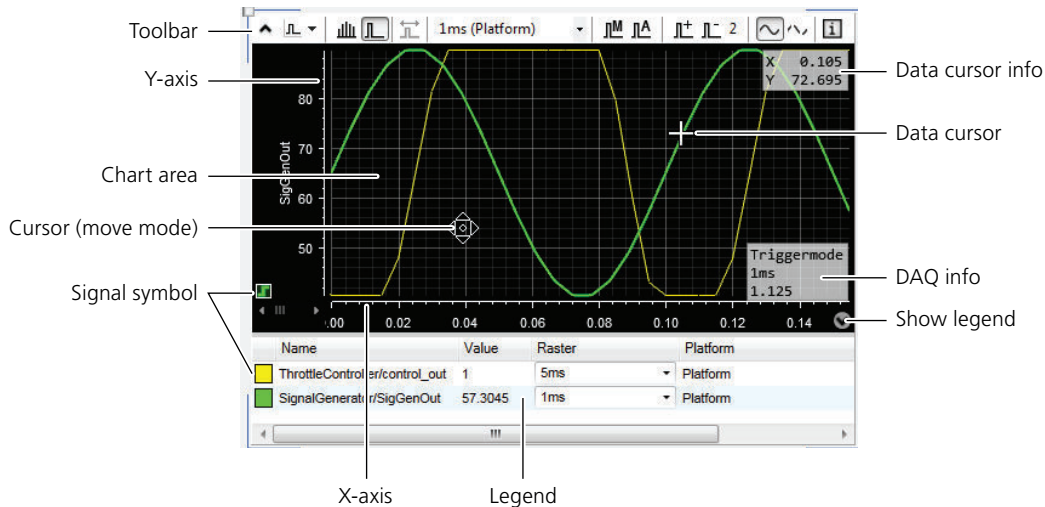
**Time Plotter**

ControlDesk's new *Time Plotter* allows you to display signals that are measured in a time-based raster (time plots).

In comparison to the standard Plotter:

- The new Time Plotter has an enhanced performance.
- The Time Plotter handling is similar to the Index Plotter (introduced with ControlDesk 5.2) and the XY Plotter.
- The fields of application of the Time Plotter, the XY Plotter, and the Index Plotter are clearly separated. This reduces the complexity in handling each plotter type.

The following illustration shows an example of the instrument:



For details, refer to *Time Plotter* ([ControlDesk Next Generation Reference](#)).

For an overview of the differences between the standard Plotter and the Time Plotter, refer to *Differences Between Plotter, Time Plotter, Index Plotter, and XY Plotter* ([ControlDesk Next Generation Basic Practices Guide](#)).

### Adding a Python script to an instrument

You can add one Python script to each instrument and assign Python code to the events of the selected instrument. This lets you extend the instrument's functionality via automation more flexibly.

The script is executed automatically each time the instrument is initialized, for example, when you open the containing layout.

The script is stored together with the instrument. So when you copy and paste the instrument, the copied instrument contains the same functionality extension as the original one.

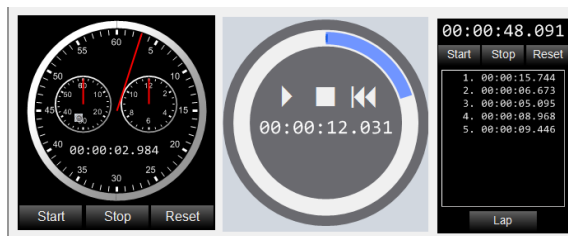
The script containing the extended functionality is maintained in the instrument even when:

- The instrument is copied and pasted.
- The instrument is added to the Custom Instruments list in ControlDesk's Instrument Selector, which lets you customize and individualize instruments.

Refer to *Adding a Python Script to an Instrument* ([📖 ControlDesk Next Generation Basic Practices Guide](#)).

### Stopwatch instruments

The Instrument Selector now provides various stopwatches.



The functionality of the stopwatches, such as starting, stopping, or displaying lap times, is added to them via instrument scripts. Refer to *Example of Adding a Python Script to an Instrument* ([📖 ControlDesk Next Generation Basic Practices Guide](#)).

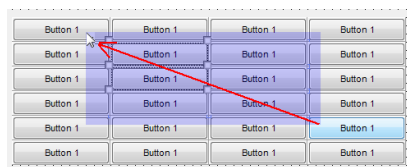
### Automating the Instrument Selector

The Instrument Selector can now be accessed via tool automation.

### Improvements to the selection of instruments

The selection of instruments on a layout has been improved.

For example, to select instruments within an instrument group, press **Ctrl+Shift** while moving the pointer over the instruments. The square symbols visualize the selected instruments:

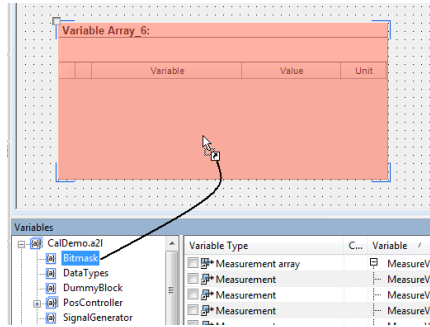


Refer to *Selecting and Positioning Instruments on a Layout* ([📖 ControlDesk Next Generation Basic Practices Guide](#)).



**Placing all the variables of a subgroup on a layout**

If an instrument is able to display all the variables of a subgroup, e.g., all the variables of a block or subsystem, you can drag the group node from the variable tree to the instrument.



## New Measurement and Recording Features (ControlDesk 5.3)

**Signal-wise loading of MDF 4.x data to handle large measurement files**

When you open an MDF 4.x file in the Measurement Data Pool, the contained signals can be loaded separately when they are requested for visualization on a layout. This lets you postprocess large MDF 4.x files.

Refer to *Measurement Files Page* ([📖 ControlDesk Next Generation Reference](#)).

**Using sample count triggers as stop triggers for time-based rasters**

A *sample count trigger* specifies the number of samples in a data capture and can now also be used as a stop trigger for a time-based raster.

Refer to *Basics on Triggers* ([📖 ControlDesk Next Generation Basic Practices Guide](#)).

## New Bus Navigator Features (ControlDesk 5.3)

---

### CAN FD support for dSPACE platforms

The Bus Navigator now supports CAN FD (CAN with Flexible Data Rate) on the following dSPACE platforms:

- DS1005
- DS1006
- DS1007
- MicroAutoBox

Refer to *Features of the Bus Navigator Specific for CAN* ([📖 ControlDesk Next Generation Advanced Practices Guide](#)).

---

### Synchronizing monitoring and logging with measurements

Monitoring and logging data can now be synchronized with measurement data, even if the measurement data is acquired from different platforms/devices.

Refer to *Bus Navigator Page* ([📖 ControlDesk Next Generation Reference](#)).

## New Data Set Management Features (ControlDesk 5.3)

---

### Handling writable measurements as parameters

You can create data sets that do not only contain parameters but also writable measurement variables.

Refer to *Adding Writable Measurement Variables to Data Sets* ([📖 ControlDesk Next Generation Basic Practices Guide](#)).

---

### Automatic data set handling on variable description reload

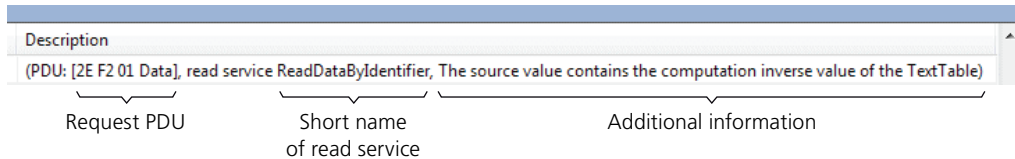
You can now specify whether to remove or to restore data sets *automatically* after reloading or replacing a variable description.

Refer to *Data Set Manager Page* ([📖 ControlDesk Next Generation Reference](#)).

## New ECU Diagnostics Features (ControlDesk 5.3)

### Enhanced description of diagnostics variables in the Variable Browser

In the Variable Browser, the description text of a diagnostics variable now provides information on the associated request PDU, the name of the read service (for parameters only), and further additional descriptive information, if available. This information is displayed in brackets. You can use it to filter the variable list.



### Customizing instrument columns

The columns of the Fault Memory Instrument and Diagnostics Instrument can now be customized. This means you can add or remove columns and change the column order.

Refer to:

- *Columns (Fault Memory Instrument)* ([📖 ControlDesk Next Generation Reference](#))
- *Columns (Diagnostics Instrument)* ([📖 ControlDesk Next Generation Reference](#))

### Fault Memory Instrument: Display of DTC number and level

The Fault Memory Instrument now also displays the number and level of DTCs.

Refer to *Fault Memory Instrument* ([📖 ControlDesk Next Generation Reference](#)).

## New Signal Editor Features (ControlDesk 5.3)

### Stimulation of V-ECU A2L variables in a VEOS simulation

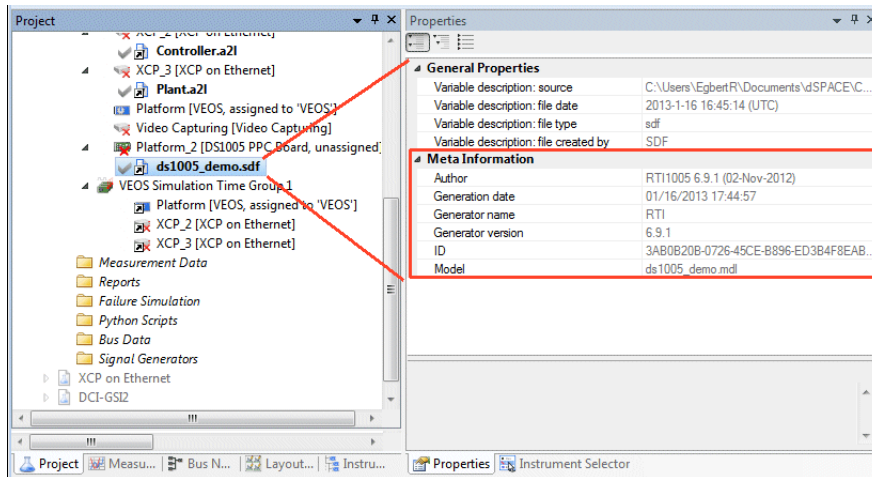
ControlDesk's Signal Editor now lets you stimulate the A2L variables of V-ECUs in an offline simulation on VEOS.

For instructions, refer to *How to Stimulate Variables of a Simulation Application* ([📖 ControlDesk Next Generation Advanced Practices Guide](#)).

## New Automation Features (ControlDesk 5.3)

### Accessing meta information of variable descriptions

You can now access meta information of variable descriptions via tool automation.



### Working with messages

ControlDesk now provides an interface to handle messages via automation. The interface lets you monitor any kind of messages such as infos, warnings or errors that are added to the log file. It also lets you filter these messages, for example, according to their severity, and react to the occurrence of specific messages via automation.

For details, refer to *Automating Message Handling* ([ControlDesk Next Generation Advanced Practices Guide](#)).



# Migrating to ControlDesk Next Generation (ControlDesk 5.3)

## Where to go from here

Information in this section

<i>Migrating to ControlDesk Next Generation (ControlDesk 5.3)</i>	93
---	----

Information in other sections

 <b>ControlDesk Next Generation Migration Guide</b> Explains migration from ControlDesk 3.x, CalDesk and prior versions of ControlDesk Next Generation to ControlDesk 5.3.	
 <b>ControlDesk Next Generation Migration of ControlDesk 3.x Automation</b> Explains migration from ControlDesk 3.x automation to ControlDesk Next Generation automation.	

## Migrating to ControlDesk Next Generation (ControlDesk 5.3)

To migrate from ControlDesk 5.2 to ControlDesk 5.3 and reuse existing experiments, you might have to carry out the following migration steps.

## Information in this topic

<i>Discontinuations in ControlDesk 5.3</i>	93
<i>Discontinuation of the CANcardX support</i>	93
<i>Discontinuation of default services according to ODX semantics for fault read functions</i>	94
<i>Migrating to ControlDesk 5.3</i>	94
<i>Changed behavior of the MultiCaptureHistoryEnabled property</i>	94
<i>Changed behavior of the Remove method (IXaMainRecorder interface)</i>	94
<i>Changes due to the support of the TextTable parameter type</i>	95
<i>Migrating from CalDesk, ControlDesk 3.x, or prior ControlDesk Next Generation 4.3 versions</i>	95

## Discontinuations in ControlDesk 5.3

**Discontinuation of the CANcardX support** Vector Informatik's CANcardX interface is no longer supported.

### **Discontinuation of default services according to ODX semantics for fault read functions**

ControlDesk no longer provides a default diagnostic service according to ODX semantics for the following ECU diagnostics functions:

- Reading fault memory entries
- Reading environment data
- Clearing single or all fault memory entries

However, ControlDesk still provides a default diagnostic service for the above functions according to protocol-specific service identification. To change from working with default services according to ODX semantics to default services according to protocol-specific service identification, you must adapt the names of the services in your automation scripts.

In addition, you can use an XML configuration file to set up services different from the default diagnostic services, see *Basics of the XML Configuration File* ([📖 ControlDesk Next Generation Advanced Practices Guide](#)).

### **Migrating to ControlDesk 5.3**

#### **Changed behavior of the**

**MultiCaptureHistoryEnabled property** The behavior of the `MultiCaptureHistoryEnabled` property of the `MeasurementTriggeredRaster / IXaMeasurementTriggeredRaster <<Interface>>` interface was incorrect in ControlDesk 5.2 and earlier. It is correct in ControlDesk 5.3 and later.

The property setting can no longer be changed during a running measurement. When you try to change the property setting during a running measurement in ControlDesk 5.3 and later, an exception occurs. You must adapt your script accordingly.

#### **Changed behavior of the Remove method**

**(IXaMainRecorder interface)** The behavior of the `Remove` method of the `MainRecorder / IXaMainRecorder <<Interface>>` interface was incorrect in ControlDesk 5.2 and earlier. It is correct in ControlDesk 5.3 and later.

A recorder can no longer be removed while it is running. When you try to remove a running recorder in ControlDesk 5.3 and later, an exception occurs. You must adapt your script accordingly.

### Changes due to the support of the TextTable parameter

**type** Up to and including ControlDesk 5.2, the TextTable type was not supported for parameters of, for example, diagnostic services. Instead, the string type was also used for parameters of the TextTable type.

As of ControlDesk 5.3, the TextTable type is supported for parameters.

As a consequence, you must adapt your automation script if the parameter type is used in the script.

---

### Migrating from CalDesk, ControlDesk 3.x, or prior ControlDesk Next Generation 4.3 versions

To migrate from CalDesk, ControlDesk 3.x, or prior ControlDesk Next Generation versions and reuse existing experiments, you might have to carry out additional migration steps. For information on the migration steps, refer to *Migrating to ControlDesk Next Generation* ([📖 ControlDesk Next Generation Migration Guide](#)).





# dSPACE HIL API .NET

## New Features of dSPACE HIL API .NET 1.7

---

### Enhanced platform support

dSPACE HIL API .NET using the MAPort implementation supports MicroLabBox.

dSPACE HIL API .NET using the MAPort implementation supports:

- MicroLabBox
- VEOS 3.3
- Stimulation for offline simulation applications with multiple environment VPU

For further information on the dSPACE HIL API .NET implementation, refer to [📖 dSPACE HIL API .NET Implementation Document](#).

---

### Discontinued EESPort support

The dSPACE HIL API .NET implementation has been adapted to the dSPACE HIL API Python Implementation that only supports the MAPort for model access. The EESPort for failure simulation has therefore been discontinued. For automating failure simulation, you can migrate to the HIL API successor, refer to *Migrating to dSPACE XIL API 2.0* on page 101.

---

### Location of the user documentation changed

The user documentation of the dSPACE HIL API implementations (.NET and Python) were moved in dSPACE HelpDesk from Software to Software - Test Automation.



---


# dSPACE Python Extensions

## New Features of dSPACE Python Extensions 1.7

---

### **dSPACE Platform Management API**

The dSPACE Platform Management API supports MicroLabBox. The platform identifier is *DS1202*.


For further information, refer to  *dSPACE Platform Management API Reference*.

---

### **Enhanced platform support**

dSPACE Python Extensions 1.7 using the HIL API Python Implementation for the MAPort supports:

- MicroLabBox
- VEOS 3.3
- Stimulation for offline simulation applications with multiple environment VPUs

For further information on the dSPACE HIL API Python Implementation, refer to  *dSPACE HIL API Python Implementation Document*.

---

### **Location of the user documentation changed**

The user documentation of the dSPACE HIL API implementations (.NET and Python) are moved in dSPACE HelpDesk from Software to Software - Test Automation.



# dSPACE XIL API

---

## Where to go from here

Information in this section

<i>New Features of dSPACE XIL API 2.0</i>	101
<i>Migrating to dSPACE XIL API 2.0</i>	101

## New Features of dSPACE XIL API 2.0

---

### Basics on dSPACE XIL API

The dSPACE XIL API is the dSPACE-specific implementation of the ASAM AE XIL API 2.0.1 standard, which is the successor of the ASAM AE HIL API. The dSPACE implementation is programmed in C#.NET and supports the Testbench providing the model access port (MAPort) and the electrical error simulation port (EESPort).

For further information on the dSPACE XIL API implementation, refer to [📖 dSPACE XIL API Implementation Guide](#).

## Migrating to dSPACE XIL API 2.0

---

### Migrating applications from dSPACE HIL API .NET to dSPACE XIL API .NET

For information on the required migration steps, refer to *Migrating HIL API Scripts to XIL API Scripts* ([📖 dSPACE XIL API Implementation Guide](#)).



# ECU Interface Manager

---

## Where to go from here

Information in this section

<i>New Features of ECU Interface Manager 1.5</i>	103
<i>Migrating to ECU Interface Manager 1.5</i>	104

---

## New Features of ECU Interface Manager 1.5

---

### Support of Renesas RH85x microcontrollers

The ECU Interface Manager now supports Renesas RH85x microcontrollers.

---

### Deleting functions

The ECU Interface Manager now lets you delete entire functions permanently, for example, to free memory space.

Deleted functions can be restored.

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

---

### Disabling the execution of functions and write accesses permanently

The ECU Interface Manager now lets you permanently disable functions and write accesses. This saves memory space.

Refer to *Insert Control Logic - Disable Execution* ([ECU Interface Manager Reference](#)).

---


### Writing back to register

For controlled write accesses, the ECU Interface Manager now lets you write the new value of a variable back to the microcontroller register to ensure data consistency.

Refer to *Insert Control Logic - Register Write Back* ([ECU Interface Manager Reference](#)).

### **Using the same service IDs for all instances**

The ECU Interface Manager now lets you use the same service IDs for all the instances of a function call or write access.

Refer to *Functions and Variables Configuration Pane* ( *ECU Interface Manager Reference*).

## Migrating to ECU Interface Manager 1.5

---

### **Migrating projects last saved with a former version of ECU Interface Manager**

In ECU Interface Manager 1.5, you can reuse projects that were last saved with a former version of the ECU Interface Manager. However, if you save them in ECU Interface Manager 1.5, you can no longer use them with a former version of the ECU Interface Manager.

**Saving a project under a new name** When you open a project that was last saved with a former version of the ECU Interface Manager in ECU Interface Manager 1.5 and save it, the ECU Interface Manager prompts you to save it under a new name. This lets you keep a version of the project that you can work with in the former version of ECU Interface Manager.



# Firmware Manager

## Features of Firmware Manager 1.2

---

### Enhanced platform support

The Firmware Manager supports the following new dSPACE hardware:

- MicroLabBox

The firmware archive `DS1202FwArchive.arc` provides the firmware components for the board.

- DS4342 CAN FD Interface Module

The firmware for the module is available in the firmware archives for:

- DS1005
- DS1006
- DS1007
- MicroAutoBox

---

### Restoring boot firmware via user interface

dSPACE platforms, except for the DS1104 R&D Controller Board, can be recognized by the Platform Manager also if they have corrupted boot firmware. You are no longer forced to use the command line utility to restore the boot firmware. This is now possible via the user interface of the Firmware Manager.

For the DS1104 R&D Controller Board, you still have to use the command line utility.



---

# dSPACE FlexRay Configuration Package

## New Features of dSPACE FlexRay Configuration Package 3.4

---

### **FlexRay Configuration Tool**

**Support of FIBEX 4.1** The FlexRay Configuration Tool now also supports FIBEX 4.1 files for describing FlexRay networks.



# Model Compare



## Product use prohibited in United States

You are not licensed to use Model Compare in the United States. You are not allowed to use or permit others to use this product in the United States or in any way that violates the laws of the United States.

### Where to go from here

Information in this section

<i>New Features of Model Compare 2.5</i>	109
<i>Migration to Model Compare 2.5</i>	110

## New Features of Model Compare 2.5

### Full-text search of hierarchy items

You can now perform a full-text search of the names of the items listed in the Reference and Comparison Hierarchies in the Model Navigator.

#### Related documentation

- *Model Navigator* ( *Model Compare Reference*)

### Display of bus comparison results

The internal hierarchy of bus signals is now displayed in the Property Inspector.

#### Related documentation

- *Property Inspector* ( *Model Compare Reference*)

---

<b>Improved display of LSB values</b>	Arbitrary LSB values are now also displayed in power-of-two notation, if possible. The power-of-two value is then displayed in parantheses, for example: 0.125 (2 <sup>-3</sup> ).
<b>Improved access to favorite settings</b>	You can now access your favorite comparison settings also via the menu bar. <b>Related documentation</b> <ul style="list-style-type: none"><li>■ <i>Favorites</i> (<a href="#">📖 Model Compare Reference</a>)</li></ul>
<b>Optional support for HIL model comparisons</b>	Model Compare now offers improved support for HIL models via dSPACE add-ons. Contact your dSPACE sales representative.

---

## Migration to Model Compare 2.5

---


<b>Save as default (button)</b>	In the Comparison Settings dialog, the Save as default button changed its label. The new label is Save as user default settings. In the favorites list, you can now select Use user default settings in addition to Use factory default settings. <b>Related documentation</b> <ul style="list-style-type: none"><li>■ <i>Save as User Default Settings</i> (<a href="#">📖 Model Compare Reference</a>)</li></ul>
<b>Line matching</b>	You can now choose between four different line matching algorithms.  The factory default value has been changed from Identical source and destination nodes and port numbers (first option and default value in Model Compare version 2.4) to Identical destination node (last option and default value in Model Compare versions ≤2.3). <b>Related documentation</b> <ul style="list-style-type: none"><li>■ <i>General Page</i> (<a href="#">📖 Model Compare Reference</a>)</li></ul>
<b>XML dumps</b>	Model Compare can create XML dump files from TargetLink models only in MIL simulation mode.

---



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

**Related documentation**

- *How to Create XML Dump Files from within MATLAB* ( *Model Compare Guide*)





# ModelDesk

## New Features of ModelDesk 4.0

---

### Processing

ModelDesk has the new Processing component. The component provides a more convenient way to parameterize simulation models on the basis of measurements than the ASMParameterization tool. This is especially useful if you want to parameterize engine models with data measured on a testbench.

The Processing component helps you to prepare the measurement data. You can adapt raw data from the measurements to the measurement data used for parameter calculation. The parameter values are calculated in MATLAB by using M functions. Then ModelDesk can write the calculated values to the parameter pages of the model.

All the data necessary for processing (e.g., measurement data, M functions) are integrated into the ModelDesk project.

---

### Tool automation

The tool automation of ModelDesk now provides the following features:

- Accessing the traffic objects, junctions, and routes of road networks.
- Starting the processing execution (see above)

---

### Road Generator

The new Road Network pane lists all the road elements and junctions of the road network. The Road and Junction panes are omitted. On the Road Network pane, you can copy and paste sections of roads which consist of several road and junction elements.

You can now import roads defined in a MATLAB MAT file.

### Parameterizing

**Automotive Simulation Models** You can parameterize the Automotive Simulation Models in this release. For details on the Automotive Simulation Models, refer to *Automotive Simulation Models (ASM)* on page 41.

**Parameter properties** The Properties pane displays the properties of a model parameter when you select it on a parameter page.

---

# MotionDesk

---

## Where to go from here

Information in this section

<i>New Features of MotionDesk 3.5</i>	115
<i>Migrating to MotionDesk 3.5</i>	116

## New Features of MotionDesk 3.5

---

### Instruments

MotionDesk now has a Multistate LED instrument that can display the value of a variable in different colors. You can specify several number ranges and their appropriate representations for a Multistate LED instrument.

---

### 3-D object library

The 3-D object library of MotionDesk contains animated characters and animals. The 3-D objects are animated independently from the simulation, but you can specify the kind of animation (e.g., walk or run) and appearance via properties. As for any other movable 3-D object, their positions can base on a data stream from the application. The new 3-D objects are protected by an extra license.

The library contains several characters such as males, females, and children, and several animals such as cows, dogs, and horses. The following illustration shows the motion of a running female.



---

**Tool automation**

You can get the name and path of the active project.

---

## Migrating to MotionDesk 3.5

---

**Migrating from MotionDesk 2.1.6 and earlier**

As of MotionDesk 2.2, there is a different project/experiment structure for managing experiment data. Earlier MotionDesk projects must therefore be migrated. For details, refer to *Migrating from MotionDesk 2.1.6 and Lower* ([📖 MotionDesk Guide](#)).

---

**Migrating from MotionDesk 2.2.1 and earlier**

Because MotionDesk now uses a new format for the 3-D objects, scenes and custom 3-D objects must be migrated. For details, refer to *Migrating from MotionDesk 2.2.1 and Lower* ([📖 MotionDesk Guide](#)).

---

**Migrating from MotionDesk 3.4**

LED instruments are migrated to Multistate LED instruments automatically.

---

# Real-Time Testing

## New Features of Real-Time Testing 2.4

---

### Supported platforms

- Real-Time Testing 2.4 supports VEOS 3.3. Now, Real-Time Testing is always enabled and can be used in more than one environment models.



# RTI/RTI-MP and RTLib

---

## Where to go from here

Information in this section

<i>New Features of RTI/RTI-MP and RTLib</i>	119
<i>Migration Aspects of RTI/RTI-MP and RTLib</i>	122

## New Features of RTI/RTI-MP and RTLib

---

### MicroLabBox

MicroLabBox is a new single-board hardware for laboratory purposes. Depending on your use cases, you can order MicroLabBox either with a *front connector panel*, which provides its analog input and output channels via two Sub-D connectors at the front of the board, or with a *top connector panel*, which provides these channels via separate BNC connectors at the top of the board.

The board can be easily connected to your host PC via Ethernet interface.

MicroLabBox provides one dual-core processor for the computation of the model (Freescale P5020) and one single-core processor for the host communication (Freescale P1011). This gives you high computation power, low I/O latency and high data throughput with data capturing. The dual-core processor allows you to use RTI-MP with MicroLabBox as a multicore platform.

MicroLabBox provides the following I/O channels:

- Analog input channels
  - 24 differential channels with a sample rate of 1 MSPS
  - 8 differential channels with a sample rate of 10 MSPS
- Analog output channels
  - 16 single-ended channels
- Digital bidirectional channels
  - 48 single-ended channels
  - 12 differential channels

In addition to the board libraries, such as the CAN Type 1 library, you can use MicroLabBox with the RTI USB Flight Recorder Blockset, which allows you to record simulation data on a USB mass storage device. With the RTI Ethernet Blockset, you can use the board's Ethernet capability for implementing Ethernet communication in your Simulink model.

For further information on the board's features, refer to [MicroLabBox Features](#).

MicroLabBox provides an FPGA module that you can program via the RTI FPGA Programming Blockset. For further information, refer to *New Features of the RTI FPGA Programming Blockset 2.8* on page 140.

### **Specific features of MicroLabBox when using RTI and**

**RTI-MP** You need to consider the following issues when using RTI and RTI-MP with MicroLabBox:

- The Data set storage option in the RTI General Build Options dialog is always set and cannot be modified.
- To identify the platform to be used for downloading the real-time application, you can use either the platform name or the network client address.
- The value of the initial simulation state is set via a parameter of the `CmdLoader` when you download the real-time application. It is not necessary to rebuild the application to change the initial simulation state. For example, it is sufficient to set the value in the Multiprocessor Setup Dialog before you start the download.
- Options that are related to the simulation state can only be set to RUN or STOP. There is no PAUSE simulation state.



**Limitations** There are the following limitations when you use RTI and RTI-MP with MicroLabBox.


- The *As fast as possible* execution mode is not supported.
- Although you can use RTI-MP with MicroLabBox, the RTI Gigalink blockset does not support MicroLabBox.
- MicroLabBox does not support the RTI Bypass Blockset.
- MicroLabBox does not support Real-Time Testing.
- MicroLabBox does not support the MotionDesk Blockset.

---

### MicroAutoBox

MicroAutoBox has some hardware and software enhancements.

**Enhanced CAN support** If you use a MicroAutoBox variant with DS1507 or DS1512, you can enhance its CAN support by using the DS4342 CAN FD Interface Module. This allows you to implement a CAN bus based on the CAN FD protocol.

For further information, refer to  MicroAutoBox Features.

**MicroAutoBox configuration tool** Some minor usability improvements have been introduced with the MicroAutoBox configuration tool `DS1401ConfigGUI.exe`.

---

### Unsupported new features of MATLAB R2014a

The following new features introduced with MATLAB R2014a are not supported by dSPACE blocksets:

- Data dictionaries
  - Simulink is able to store design data that your model uses in a data dictionary as a persistent repository.

---

### Unsupported new features of MATLAB R2014b

The following new features introduced with MATLAB R2014b are not supported by dSPACE blocksets:

- Simulink Function Subsystems
  - You cannot use dSPACE blocks, such as RTI and RTI-MP blocks (except for the Default CPU block) or blocks from board-specific RTI blocksets, in a Simulink Function Subsystem. The contents of Simulink Function Subsystem is not generated into the TRC file.
- Model Templates
  - You cannot use dSPACE blocks, such as RTI and RTI-MP blocks or blocks from board-specific RTI blocksets, in a model template.

---

### Updated DS1006 GNU C/C++ Compiler

The compiler toolchain for the DS1006 Processor Board has been updated to improve performance, stability and compatibility.

The new features are:

- GNU compiler version 4.8.3
- The toolchain includes a port of the Newlib C library for embedded systems which provides standard C library functionality.
- The toolchain now supports C++ for custom code. The installation of the separate C++ Integration Kit is no longer required.



The code generated from the model by RTI is C code. The C++ setting for the Language option in the Code Generation page is not supported.

- The tool chain now supports C++ exception handling and RTTI. However, it is recommended not to use these features because of performance and timing reasons.  
Use exception handling only for the detection of critical errors. The real-time application should be terminated when an exception occurs.

#### Updated Microtec PowerPC C/C++ compiler

The compiler toolchain for the PowerPC boards (DS1005, DS1103, DS1104, MicroAutoBox) has been updated to provide C++ support for custom code. The installation of the separate C++ Integration Kit is no longer required.



The code generated from the model by RTI is C code. The C++ setting for the Language option in the Code Generation page is not supported.

## Migration Aspects of RTI/RTI-MP and RTLib

#### Modifications concerning the TRC file generation

The following settings in the RTI variable description file options page of the Code Generation Dialog have been changed:

- The Apply subsystem read/write permissions setting is no longer available.
- The following settings are now cleared by default:
  - Include signal labels
  - Include virtual blocks

The new default values are only relevant for new models or if you switch your model to another platform.

- The Include only Simulink.Parameter and Simulink.Signal objects with global storage class setting now also considers the BusSystems group of the RTI CAN MultiMessage Blockset and the RTI LIN MultiMessage Blockset.

---

### Handling of Bus Selector block changed

With MATLAB R2014a, the Simulink Bus Selector block has been virtualized. In combination with dSPACE Release 2014-B, the outputs of this block and other virtual blocks that are connected to it are not generated into the variable description, if the inport of the Bus Selector block is connected to a *non-virtual bus*.

To access a signal of a bus, do the following:

- Access the signal directly from the bus,  
or
- Connect a non-virtual block to the signal, for example, a Gain block with 1 as the factor. Its output is available in the variable description.

If the inport of the Bus Selector block is connected via virtual bus, the outputs are generated into the variable description file without applying special methods.

---

### Generated code from Simulink Coder has changed

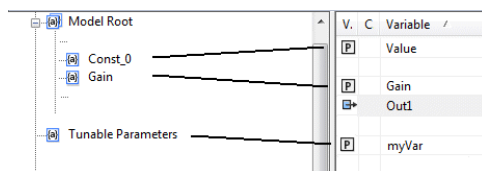
**Using MATLAB R2014a** With the MATLAB/Simulink R2014a release, code generation by Simulink Coder has changed. Therefore, also the generation of variable descriptions (TRC files) in RTI has changed.

You have to use the `revertInlineParametersOffToR2013b` command before you start a build process. This command enables the Simulink Coder behavior and dSPACE TRC file generation in MATLAB R2014a similar to that in MATLAB R2013b and before. The `revertInlineParametersOffToR2013b` command is shipped with MATLAB R2014a as part of the Simulink Coder product.

For more details, see the Simulink Coder R2014a release notes (*Simpler behavior for tuning all parameters and support for referenced models*): <http://www.mathworks.de/de/help/rtw/release-notes.html> (MathWorks Account required).

**Using MATLAB R2014b** The modification in the Simulink Coder affects the entries in the generated TRC file only if you have the Inline Parameters optimization option cleared. Workspace variables that are referenced by one or more block parameters are now handled as global parameters in the model and not as local parameters of the block. They are stored to the Tunable Parameters group in the generated TRC file. If you now change the value of such a global parameter, any block parameter referencing this global parameter changes as well. The behavior of your simulation will change.

Example:



In the model, you have configured the Const\_0 block parameter and the Gain block parameter by using the workspace variable `myVar`. Independently of the Inline parameters option on the Optimization page, the workspace variable is generated into the variable description. The variables can be used in ControlDesk Next Generation, as before, by connecting Const\_0/Value and Gain/Gain to instruments, but internally, the parameter values depend on the value of `myVar`. If you modify one of the parameters, all others are changed too.

Workspace variables are generated as local block parameters in the following cases:

- The parameter is a structure or a structure item, for example, `myStruct.Value`.
- The parameter is used in an expression or in a function, for example, `myValue + 1` or `sin(myValue)`.

Local block parameters are also generated if references to mask parameters are used. The mask parameter itself is not generated into the variable description.

### Migrating DS1006 real-time applications

Because of the updated DS1006 GNU C/C++ compiler, it is recommended to rebuild your real-time applications. While objects and libraries referencing RTLib functionality are compatible, objects and libraries referencing standard C/C++ functionality might conflict with the updated C and C++ libraries.

# RTI Bypass Blockset

---

## Where to go from here

Information in this section

<i>New Features of the RTI Bypass Blockset 3.3</i>	125
<i>Migrating to RTI Bypass Blockset 3.3</i>	127

---

## New Features of the RTI Bypass Blockset 3.3

---

### RTI Bypass Blockset

**Support of MATLAB x64** The RTI Bypass Blockset now also supports 64-bit MATLAB versions.

**Improved assignment of FlexRay frames to FlexRay buffers** The RTI Bypass Blockset comes with the following improvements concerning the mapping of FlexRay frames to FlexRay buffers:

- The **Buffers Configuration** page of the Setup block provides mapping information for FlexRay buffers. You can open a dialog that shows the buffers that can be assigned to each available LPDU. If no buffer can be assigned, you get information on the reason.  
Displaying this mapping information can be useful, for example, to investigate why a certain LPDU cannot be assigned.
- You can use the same frame name for different XCP LPDUs.

Refer to *Buffers Configuration Page (RTIBYPASS\_SETUP\_BLx for XCP on FlexRay)* ([RTI Bypass Blockset Reference](#)).

**Labeling the ECU application's binary content** When building internal bypass code, an extended database file (output A2L file) and a merged ECU application (output ECU application) are generated. The RTI Bypass Blockset now lets you specify a label for the ECU application's content and integrate it into the exported binary code. You can use predefined macros to define the labels dynamically. For example, you can include the product, the user, and the file creation date and time. Refer to *Build Page (RTIBYPASS\_SETUP\_BLx for INTERNAL)* ([RTI Bypass Blockset Reference](#)).

The memory address and the label to be used can be specified by the A2L file. dSPACE provides the new AML file `dSPACE_INTERNAL_Bypass_v1_3_0.aml` for the `IF_DATA` `dSPACE_INTERNAL_BYPASS` entry in this context. Refer to *Interface Description Data for Internal Bypassing* ([Interface Description Data Reference](#)).

**Support of XCP 1.2** The RTI Bypass Blockset supports A2L files containing XCP-specific `IF_DATA` entries based on the XCP 1.2 standard. The supported features are the same as for XCP 1.1.

---

### RTI Bypass Blockset MATLAB API

**Support of enhancements to RTI Bypass Blockset** The RTI Bypass Blockset MATLAB API supports the enhancements to the RTI Bypass Blockset.

Refer to the [RTI Bypass Blockset MATLAB API Reference](#).

## Migrating to RTI Bypass Blockset 3.3

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

The current release contains RTI Bypass Blockset 3.3, 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 compared to these prior RTI Bypass Blockset versions. If you have a Simulink model built with RTI Bypass Blockset 2.5 or earlier and open it with RTI Bypass Blockset 3.3, the old data dictionary file (with file name extension .dd) is replaced by a new data dictionary file (.vdb) using the information stored in the Setup block as soon as you open and close the Setup block dialog via OK, or open the Read, Write, Upload or Download block dialog and click the Fill Variable Selector button on the Variables page.

If you have a model that was saved with RTI Bypass Blockset 3.3 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 as soon as you update the A2L files in the Setup block or open the Read, Write, Upload or Download block and click the Fill Variable Selector button on the Variables page. The data dictionary file created under RTI Bypass Blockset 3.3 (\*.vdb) remains on disk.

To enable the RTI Bypass Blockset to recreate the data dictionary, the database files specified in the Setup block must be accessible at the specified location and must be unchanged.

- *Working with models from RTI Bypass Blockset 2.6 up to and including RTI Bypass Blockset 3.2:* If you have a Simulink model built with RTI Bypass Blockset 2.6 up to RTI Bypass Blockset 3.2 and open it with RTI Bypass Blockset 3.3, 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 use the model with RTI Bypass Blockset 2.6 up to RTI Bypass Blockset 3.2 again, 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 Blockset

## New Features of the RTI CAN Blockset 3.3

---

### **New supported platform**

The RTI CAN Blockset supports MicroLabBox, which provides 2 CAN channels.

The blockset supports *CAN partial networking* for MicroLabBox if the ISO11898-6 transceiver type is selected.

Refer to *Partial Networking Page (RTICAN CONTROLLER SETUP)* ([📄 RTI CAN Reference](#)).



---

# RTI CAN MultiMessage Blockset

---

## Where to go from here

Information in this section

<i>New Features of the RTI CAN MultiMessage Blockset 4.0</i>	131
<i>Migrating to RTI CAN MultiMessage Blockset 4.0</i>	132

---

## New Features of the RTI CAN MultiMessage Blockset 4.0

---

### New supported platform

The RTI CAN MultiMessage Blockset supports MicroLabBox. MicroLabBox provides two CAN channels.

**CAN partial networking** The blockset supports CAN partial networking for MicroLabBox if the ISO11898-6 transceiver type is used. Refer to *Partial Networking Page (RTICANMM ControllerSetup)* ([📖 RTI CAN MultiMessage Blockset Reference](#)).

---

### Support of FIBEX 4.1

The RTI CAN MultiMessage Blockset now also supports FIBEX 4.1 files as database files.

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

### Support of CAN FD protocol

The RTI CAN MultiMessage Blockset now also supports the CAN FD (*CAN with Flexible Data Rate*) protocol. Compared with the classic CAN protocol, CAN FD comes with an increased bandwidth for the serial communication. The improvement is based on two factors:

- CAN messages with longer data fields (up to 64 bytes)
- An optional higher bit rate

Applications can take advantage of one or both factors.

Refer to *Basics on Working with CAN FD* ([RTI CAN MultiMessage Blockset Reference](#)).

### RTI CAN MultiMessage Blockset Tutorial

The RTI CAN MultiMessage Blockset Tutorial now provides a lesson on using end-to-end communication protection (E2E protection) parameters in checksum algorithms. It explains how to use E2E protection parameters from the database file to implement checksum algorithms based on E2E protection parameters in a real-time application.

Refer to *Lesson 15 (Advanced): Using Checksums Based on E2E Protection Parameters* ([RTI CAN MultiMessage Blockset Tutorial](#)).

## Migrating to RTI CAN MultiMessage Blockset 4.0

### Working with models from earlier RTI CAN MultiMessage Blockset versions

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

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

- In the MATLAB Command Window, enter `rtimmsu_update('System', gcs)`.  
For further 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 further information, refer to *Limitations with RTICANMM* ([RTI CAN MultiMessage Blockset Reference](#)).

---

### **Compiler messages when using code generated by an RTI CAN MultiMessage Blockset version < 4.0**

If you use code that was generated by an RTI CAN MultiMessage Blockset version < 4.0, several compiler warning messages containing the phrase <<argument of type "can\_tpl\_canChannel \*" is incompatible with parameter of type "DsTCanCh">> will appear during the build process of your simulation model due to a modified data type. These warnings can be ignored and will disappear after re-generation of the RTICANMM code by using the current blockset version.

---

### **Using existing checksum algorithms**

Checksum algorithms originally developed for an application containing CAN messages cannot be reused for applications containing CAN FD messages, because CAN FD includes new message types and longer data fields. Existing checksum algorithms can still be used for applications containing classic CAN messages only. For CAN FD applications, you must adapt the checksum algorithms.



---

# RTI Electric Motor Control Blockset

## Features of RTI Electric Motor Control Blockset 1.0

---

### **New blockset**

The RTI Electric Motor Control Blockset is a new blockset providing special functions to implement controllers for state-of-the-art electric motors.

It provides blocks for:

- Multi-channel PWM signal generation
- Block-commutated PWM signal generation
- Encoder
- HALL sensor

The blockset supports MicroLabBox.

For further information, refer to  *RTI Electric Motor Control Blockset Reference*.





---

# RTI Ethernet Blockset

## New Features of the RTI Ethernet Blockset 1.1

---

### **New platform support**

The RTI Ethernet Blockset supports MicroLabBox.

---

### **Configurable switch behavior**

The DS1007 PPC Processor Board and MicroLabBox provide three Ethernet connectors which are internally connected to an Ethernet switch. The browser-based configuration and management tools of these boards now provide the new **Switch Configuration** page that allows you to specify the behavior of the switch. By this, you can select which Ethernet connector(s) can be used for I/O device communication or host PC communication.

For further information, refer to *Basics on Browser-Based Configuration and Management Tool* ([📖](#) *DS1007 Hardware Installation and Configuration Guide*).

---

### **Documentation of the RTLib functions available**

The RTLib functions used by the RTI Ethernet Blockset are now documented in the *DSIOETH RTLib Reference*. This allows you to handcode communication via the I/O Ethernet interface of the DS1007 PPC Processor Board or MicroLabBox. For further information, refer to [📖](#) *DSIOETH RTLib Reference*.



---

# RTI FPGA Programming Blockset

---

**Where to go from here**

Information in this section

<i>New Features of the RTI FPGA Programming Blockset 2.8</i>	140
<i>Migrating to RTI FPGA Programming Blockset 2.8</i>	141

## New Features of the RTI FPGA Programming Blockset 2.8

### Extended Xilinx® support

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

Xilinx Design Tools Version <sup>1)</sup>	Operating System	MATLAB Version
ISE 14.7 <sup>2)</sup> (32-bit version)	<ul style="list-style-type: none"> <li>Windows 7 Business, Ultimate, and Enterprise SP1 (32-bit version and 64-bit version)</li> </ul>	32-bit and 64-bit versions of: <ul style="list-style-type: none"> <li>MATLAB R2013a</li> <li>MATLAB R2013b</li> </ul>
ISE 14.7 <sup>2)</sup> (64-bit version)	<ul style="list-style-type: none"> <li>Windows 7 Business, Ultimate, and Enterprise SP1 (64-bit version)</li> </ul>	32-bit and 64-bit versions of: <ul style="list-style-type: none"> <li>MATLAB R2013a</li> <li>MATLAB R2013b</li> </ul>
Vivado 2014.2 <sup>3)</sup> (32-bit version)	<ul style="list-style-type: none"> <li>Windows 7 Business, Ultimate, and Enterprise SP1 (32-bit version)</li> </ul>	32-bit versions of: <ul style="list-style-type: none"> <li>MATLAB R2013a</li> <li>MATLAB R2013b</li> <li>MATLAB R2014a</li> </ul>
Vivado 2014.2 <sup>3)</sup> (64-bit version)	<ul style="list-style-type: none"> <li>Windows 7 Business, Ultimate, and Enterprise SP1 (64-bit version)</li> </ul>	64-bit versions of: <ul style="list-style-type: none"> <li>MATLAB R2013a</li> <li>MATLAB R2013b</li> <li>MATLAB R2014a</li> </ul>

<sup>1)</sup> Note that due to the introduction of the Vivado® software, Xilinx® no longer supports the Xilinx System Generator for DSP in combination with the ISE Design Suite after MathWorks® MATLAB® and Simulink® Release R2013b.

<sup>2)</sup> The DS5203 FPGA Board (LX50) and the DS2655 FPGA Base Board also support the WebPACK Editions of the Xilinx design tools.

<sup>3)</sup> Only supported by MicroLabBox.

### General enhancements

**New platform support** The RTI FPGA Programming Blockset now provides a framework for MicroLabBox.

The platform identifier of MicroLabBox is DS1202. The board also provides the required I/O connectors with signal conditioning. The framework for MicroLabBox is therefore named *DS1202 with onboard I/O*.

The framework provides the following features:

- A/D conversion
- D/A conversion
- Digital I/O
- Serial Interface (RS232 and RS422/485)
- State information for the FPGA application and the processor application
- Access to the programmable LEDs of the board

**Enhanced channel name setting** The Channel name setting for the FPGA\_XDATA blocks is now also available for the FPGA\_IO\_READ\_BLx and FPGA\_IO\_WRITE\_BLx blocks. This setting can be used to enter a user-specific channel name instead of a generated one. It allows you to specify a more descriptive or a shorter name.

For further information, refer to  *RTI FPGA Programming Blockset - FPGA Interface Reference*.

**Changed board name** The DS2655 FPGA Base *Module* is renamed DS2655 FPGA Base *Board*.

**Simplified handling of ConfigurationDesk custom functions** The FPGA applications that you built for a SCALEXIO system containing a DS2655 FPGA Base Board and one or more DS2655 Multi-I/O Modules, can now be imported to ConfigurationDesk more easily than before. For further information, refer to *ConfigurationDesk – Implementation* on page 74.

#### Related topics

Basics

- *Migrating to RTI FPGA Programming Blockset 2.8* on page 141

## Migrating to RTI FPGA Programming Blockset 2.8

#### Objective

There are different methods to migrate an existing model, depending on the blockset version used.

#### Migrating from RTI FPGA Programming Blockset 1.0 to 2.8

Because the RTI FPGA Programming Blockset 1.0 (released with dSPACE Release 6.4) was not fully implemented, a model that you implemented with it must be migrated manually. You must replace each block of the RTI FPGA Programming Blockset by a new one to make the model compatible with the current dSPACE RTI environment for modeling, building and executing.



The update function of the script interface does not support RTI FPGA Programming Blockset 1.0.

### Migrating from RTI FPGA Programming Blockset 1.1 and higher to 2.8

If you have implemented your FPGA application using RTI FPGA Programming Blockset Version 1.1 and later, and want to use it with RTI FPGA Programming Blockset 2.8, you must update the FPGA framework. You can use the script interface for this, refer to *Updating the FPGA framework using the script interface* on page 142.

You also have to update the framework if you have updated from MATLAB R2008b or earlier to MATLAB R2011b or later.

### Updating the FPGA framework using the script interface



It is recommended to backup your model before starting migration.

The script interface provides the `FPGAFrameworkUpdate` method to update a framework. You can decide whether to set the block parameters to their initial values or leave them unchanged.

#### To update the FPGA framework without changing the values of the block parameters

```
rtifpga_scriptinterface('FPGAFrameworkUpdate',
    <SimulinkHandle>)
```

The script handles all the subsystems in the model/subsystem which is specified by the Simulink handle. The parameters of the blocks are unchanged after updating to the current framework version.

Example: The following script updates the FPGA framework for any FPGA subsystem in the processor model called *MyProcModel*. The specified values of the block parameters are not changed.

```
ProcModelHandle = get_param('MyProcModel','handle')
rtifpga_scriptinterface('FPGAFrameworkUpdate',
    ProcModelHandle)
```

#### To update the FPGA framework and reset the values of the block parameters to their initial values

```
rtifpga_scriptinterface('FPGAFrameworkUpdate',
    <SimulinkHandle>, 'ReInit')
```

The script handles all the subsystems in the model/subsystem which is specified by the Simulink handle. The parameters of the blocks are reset to their initial values after updating to the current framework version.

```
ProcModelHandle = get_param('MyProcModel','handle')
rtifpga_scriptinterface('FPGAFrameworkUpdate',
    ProcModelHandle, 'ReInit')
```

**ConfigurationDesk custom functions incompatible with dSPACE Release 2014-B**



Relevant for SCALEXIO systems with a DS2655 FPGA Base Board and a DS2655M1 I/O Module

A custom function generated by using RTI FPGA Programming Blockset 2.5 from dSPACE Release 2013-A and the real-time applications (\*.RTA) containing the custom function are incompatible with dSPACE Release 2014-B. The FPGA model has to be rebuilt by using RTI FPGA Programming Blockset 2.8 from dSPACE Release 2014-B to produce a usable custom function.





---

# RTI LIN MultiMessage Blockset

---

## Where to go from here

Information in this section


<i>New Features of the RTI LIN MultiMessage Blockset 2.4</i>	145
<i>Migrating to RTI LIN MultiMessage Blockset 2.4</i>	145

## New Features of the RTI LIN MultiMessage Blockset 2.4

---

### Support of FIBEX 4.1

The RTI LIN MultiMessage Blockset now also supports FIBEX 4.1 files as database files.

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

## Migrating to RTI LIN MultiMessage Blockset 2.4

---

### Working with models from earlier RTI LIN MultiMessage Blockset versions

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

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

- In the MATLAB Command Window, enter `rtimmsu_update('System', gcs)`.  
For further 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 further information, refer to *Limitations of RTI LIN MultiMessage Blockset* ([📖 RTI LIN MultiMessage Blockset Reference](#)).

---

# RTI USB Flight Recorder Blockset

## New Features of the RTI USB Flight Recorder Blockset 1.2

---

### Enhanced platform support

The RTI USB Flight Recorder Blockset supports MicroLabBox.

For further information, refer to [📖 RTI USB Flight Recorder Blockset Reference](#).

---

### Separate Documentation of the RTLib functions

The RTLib functions that you must use for handcoding the USB flight recording feature are now documented in the *USB Flight Recorder RTLib Reference*. Previously, it was integrated in the RTLib references of the supported platforms.

For further information, refer to [📖 USB Flight Recorder RTLib Reference](#).



---

# SCALEXIO Firmware

## New Features of the SCALEXIO Firmware 3.1

---

### SCALEXIO Processing Unit

The SCALEXIO firmware supports a new real-time PC. For details, refer to *SCALEXIO Real-Time PC Data Sheet* ([PDF](#) *SCALEXIO Hardware Installation and Configuration*).

---

### I/O functions

The functions for SENT In, Lambda DCR, and Lambda NCCR have been improved, see *New Features of ConfigurationDesk 5.2 (Implementation Version)* on page 74.



---

# SystemDesk

---

**Where to go from here**

Information in this section

<i>New Features of SystemDesk 4.3</i>	152
<i>Migrating to SystemDesk 4.3</i>	156

# New Features of SystemDesk 4.3

---

**Where to go from here**

Information in this section

<i>New General Features</i>	152
<i>Modeling Software Architectures</i>	153
<i>Modeling Systems</i>	154
<i>Configuring ECUs</i>	154
<i>Validating Elements</i>	155

## New General Features

---

SystemDesk 4.3 has the following new general features.

---

**AUTOSAR Releases supported by SystemDesk 4.3**

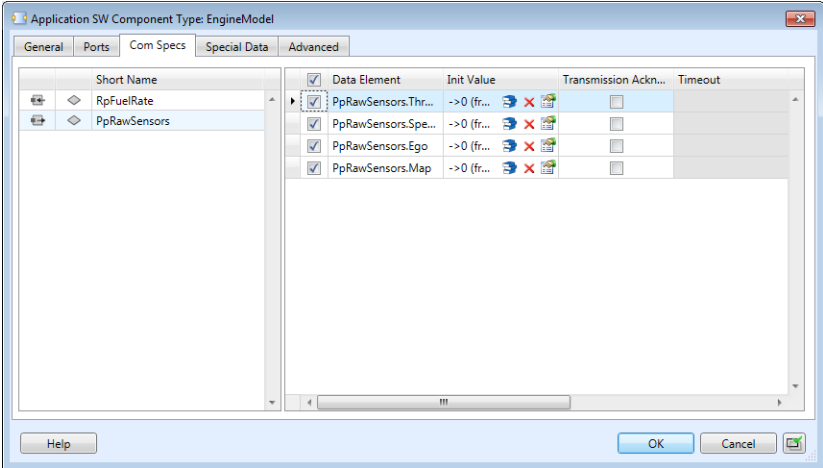
SystemDesk 4.3 supports the modeling of software and system architectures according to AUTOSAR 4.1.3, 4.1.2, 4.1.1, 4.0.3, and 4.0.2.



# Modeling Software Architectures

## Improved handling of communication specifications

SystemDesk now lets you view and edit communication specifications at a glance for all the ports of a software component. So you can edit init values for multiple data elements and ports simultaneously.



## Modeling Systems

### Improved mapping of inter-ECU communication to network communication

SystemDesk now provides the Data Mapping Editor for conveniently mapping inter-ECU communication to system signals representing network communication. The illustration below shows the editor for a demo project.

Source		Target				Data Mapping		
ECU(s)	SWC(s)	Port(s)	Element(s)	Element(s)	Port(s)	SWC(s)	ECU(s)	System Signal (Group)
EngineModel	EngineModel	PpRawSensors	Throttle	Throttle	RpRawSensors	FuelsysSensors	Controller	RawThrottle
EngineModel	EngineModel	PpRawSensors	Speed	Speed	RpRawSensors	FuelsysSensors	Controller	RawSpeed
EngineModel	EngineModel	PpRawSensors	Ego	Ego	RpRawSensors	FuelsysSensors	Controller	RawEgo
EngineModel	EngineModel	PpRawSensors	Map	Map	RpRawSensors	FuelsysSensors	Controller	RawMap
Controller	FuelsysSensors	PpCorrectedSensors	Sensors	Sensors	RpCombiSensors	FuelsysCombi	Combi	CorrectedSensors
Controller	FuelsysController	PpFuelRate	FuelRate	FuelRate FuelRate	RpFuelRate RpFuelRate	FuelsysCombi EngineModel	Combi EngineModel	FuelRate

## Configuring ECUs

### Generating V-ECU implementations and V-ECUs for systems with LIN network clusters

SystemDesk now supports generating V-ECU implementations and V-ECUs for systems with LIN network clusters.

You can import LDF files or AUTOSAR ARXML files with LIN communication elements.

**V-ECU implementations** SystemDesk lets you generate basic software module configurations for *LinIf*, *PduR*, and *COM* modules. In addition, SystemDesk generates the C code for *PduR* and *COM*.

**V-ECUs** SystemDesk lets you generate V-ECUs for VEOS simulation.

## Validating Elements

### Validation results

SystemDesk now shows validation results as in the illustration below.

Validation results are clearly visualized. You can now repeat a validation or save validation results to a file. This allows you to exchange validation results or keep them as project documentation.

Validation of **Communication** failed with 1 errors and 1 warnings.  
 Rule configuration: C:\Users\HendrikK\AppData\Local\ldSPACE\SystemDesk.exe\_StrongName\_onfswsym3m1vucyglm0ls0c1dij11duf4.3.0.0\ValidationRuleConfiguration.xml  
 Context: None  
 55 internal and 0 custom rules applied, 166 project elements examined.

System or ECU configuration elements were validated without a context. When validating these elements, a system or ECU configuration should be set as the context. Otherwise, some validation rules might not report all problems.

Select Context and Restart Validation | Do Not Show This Message Again

Message	Element Path	Rule
Problem Domain: General 1 Messages		
<b>SystemSignals</b> - Two or more elements within this namespace have the same short name 'Upi'. References might not be resolved correctly.	/Communication/SystemSignals	R101
Problem Domain: Software System 1 Messages		
<b>Upi</b> - Found an empty UUID. This can lead to problems in exchanging AUTOSAR elements in a tool chain.	/Communication/SystemSignals/Upi	R431

1 errors (0 filtered), 1 warnings (0 filtered) | [Save Layout](#) | [Restore Default Layout](#) | Updated: 13:29:58

# Migrating to SystemDesk 4.3

## Migrating to SystemDesk 4.3

---

SystemDesk 4.3 automatically migrates SystemDesk 4.1 and 4.2 SDP project files upon loading.



You are recommended to install the most recent patch for SystemDesk 4.1 or 4.2. Then, save the SDP project files you want to migrate before opening them in SystemDesk 4.3.

---

# TargetLink

---

**Where to go from here**

Information in this section

<i>New Features of TargetLink 4.0 and TargetLink Data Dictionary 4.0</i>	158
<i>Migrating to TargetLink 4.0 and TargetLink Data Dictionary 4.0</i>	184
<i>Changes in Future TargetLink Versions</i>	218

# New Features of TargetLink 4.0 and TargetLink Data Dictionary 4.0

## Where to go from here

Information in this section

<i>Modeling in Simulink or Stateflow</i>	158
<i>Code Generation Core Functionality</i>	164
<i>Data Dictionary and Data Management</i>	167
<i>AUTOSAR</i>	172
<i>Test Support</i>	175
<i>Code Generator Options</i>	177
<i>Tool Chain Integration</i>	179
<i>Other</i>	180
<i>API Commands</i>	181

## Modeling in Simulink or Stateflow

## Where to go from here

Information in this section

<i>Support of Matrix Signals</i>	159
<i>Newly Supported Simulink Blocks</i>	159
<i>Improved Bus Support</i>	160
<i>Dynamic Look-Up Tables</i>	160
<i>Improvements to TargetLink's Simulation Frame</i>	161
<i>Improvements to Scaling-Invariant Systems</i>	162
<i>Additionally Supported Block Properties</i>	162
<i>Central Specification of Function Subsystem Signatures</i>	163

## Support of Matrix Signals

### 2-D matrix signals

TargetLink supports code generation for 2-D signals and parameters (→ Matrix signal) and lets you perform simulations in MIL/SIL/PIL simulation mode.



For RTOS code generation mode, 2-D signals are not allowed to pass task boundaries. Within tasks, 2-D signals are supported.

#### Related documentation

- *Introduction to Working With Matrix Signals* ([📖 TargetLink Preparation and Simulation Guide](#))
- *Code Pattern for Vectors and Matrices* ([📖 TargetLink Preparation and Simulation Guide](#))
- *Examples of Working With Matrix Signals* ([📖 TargetLink Preparation and Simulation Guide](#))
- *Blocks not supporting matrix signals* ([📖 TargetLink Orientation and Overview Guide](#))

## Newly Supported Simulink Blocks

TargetLink now supports the following Simulink blocks:

- Matrix Concatenate
- Permute Dimensions
- Reshape

#### Related documentation

- *Supported Simulink Blocks* ([📖 TargetLink Block and Object Reference](#))
- *Working With Matrix Signals* ([📖 TargetLink Preparation and Simulation Guide](#))

## Improved Bus Support

### Mapping Entire Buses to Predefined Structs

This TargetLink version lets you create a predefined struct type in the TargetLink Data Dictionary (DD), and assign a complete bus to this DD struct type in TargetLink's BusInport and BusOutport blocks. This is especially helpful if the bus consists of many bus elements, and/or the type or variable is used several times in a model or across models.

In addition, you can reference DD struct types or struct variables in bus-supporting blocks, for example Switch, Multiport Switch, Merge, and Unit Delay blocks.

#### Related documentation

- *Basics on the Representation of Buses in the Production Code* ([📖 TargetLink Customization and Optimization Guide](#))
- *Mapping Entire Buses to Explicit Structure Variables and Type Definitions in the Code* ([📖 TargetLink Customization and Optimization Guide](#))

## Dynamic Look-Up Tables

### Changing table data during simulation

For the three TargetLink look-up table blocks listed below, you can specify the table data not only in the block dialog but you can also feed the table data (1-D or 2-D) into the block through the block's optional *table data input port*. The latter allows you to change the table data during simulation and run time of the generated code.

- Direct Look-Up Table (n-D)
- Prelookup
- Interpolation Using Prelookup

#### Related documentation

- *Principles on Look-up Tables* ([📖 TargetLink Preparation and Simulation Guide](#))
- *How to Prepare Dynamic Look-Up Table Specification* ([📖 TargetLink Preparation and Simulation Guide](#))



## Improvements to TargetLink's Simulation Frame

---

When switching to SIL or PIL simulation mode, TargetLink now disables the MIL subsystem by setting the Commented block parameter to `on`.

This is beneficial in the following ways:

- Improved speed of model initialization in SIL or PIL simulation mode
- Improved speed when switching to SIL or PIL simulation mode

### Compatibility consideration

- Simulink does not perform consistency checks for commented blocks.
- By default, Simulink's `find_system` API function does not include commented blocks in its search.
- Simulink does not allow using OpenFcn callbacks that directly or indirectly change the Commented block property. For details, refer to *Various Migration Aspects* on page 208.

**Solution** Two possible solutions exist:

- Adapt your user scripts so that the simulation mode is switched to MIL first.
- Adapt your user scripts to using TargetLink API functions:
  - `tl_get_blocks`
  - `tl_get_sfojects`
  - `tl_find`

The new **Activate MIL** entry in the TargetLink menu in Simulink Model windows lets you easily switch into MIL simulation mode.

## Improvements to Scaling-Invariant Systems

### Improved scaling inheritance

For scaling-invariant subsystems, you can optionally specify the inport(s) that influence an outport's scaling. You can specify the port mapping in the scaling propagation function. The Code Generator takes this mapping into account during code generation if the (new) `UtilizeExplicitDependenciesForScalingInvariantSystems` Code Generator option is enabled. The feature is beneficial for a better scaling propagation in case of loops including scaling-invariant systems. Dependencies of output scalings with regard to individual inputs are now taken into account, which helps to resolve loops during scaling-propagation.

#### Related documentation

- *Details on the Scaling Propagation Function* ([📖 TargetLink Customization and Optimization Guide](#))

### Access to System Handle

Scaling propagation functions now also provide access to the handle of the instance of the scaling-invariant system. This is particularly helpful if the scaling depends not only on inputs but also on instance-specific block data, e.g. specified in mask variables.

## Additionally Supported Block Properties

### Additional options for the Assignment block

The TargetLink Assignment block now supports the `Starting index (dialog)` and `Starting index (port)` options.

#### Related documentation

- *Output Page (Assignment Block)* ([📖 TargetLink Block and Object Reference](#))

### Additional options for the Selector block

TargetLink now supports the `Starting index (dialog)` and `Starting index (port)` options of the Simulink Selector block.

### Improved plotting for the Sink block

The TargetLink Sink block now supports the specification of individual plot channels rather than a general switch for all plot channels at once.

#### Related documentation

- *Logging Page (Sink Block)* ([📖 TargetLink Block and Object Reference](#))

**Date property for Stateflow data**

TargetLink now provides a Date property for Stateflow data just as for TargetLink blocks. The property helps you to identify the latest changes, e.g., when inspecting Stateflow objects by using the TargetLink Property Manager

## Central Specification of Function Subsystem Signatures

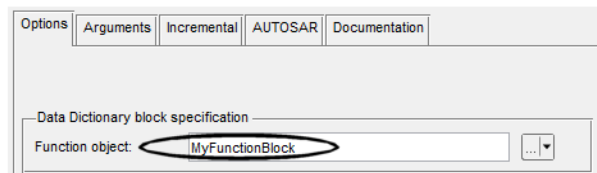
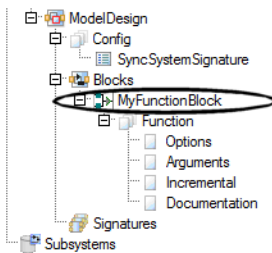
You can now centrally specify the signature of a function subsystem in the Data Dictionary. Additionally, you can generate function subsystems from these central specifications:

**Information in this topic**

<i>DD Function Block object specifying Function Block data</i>	163
<i>DD Signature object specifying a subsystem's signature</i>	164

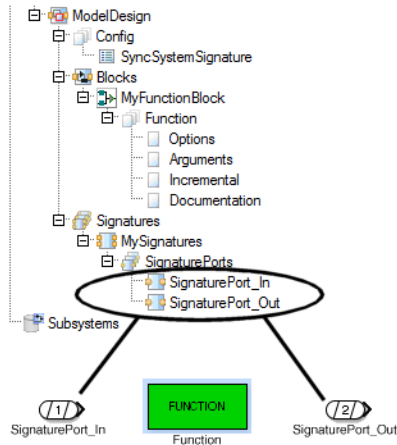
**DD Function Block object specifying Function Block data**

You can now specify Function Block data from within the TargetLink Data Dictionary by using a Function Block object. You can then reference this object in a TargetLink Function Block as shown below.



### DD Signature object specifying a subsystem's signature

You can also specify a subsystem's signature by using a Signature object. To do so, use the Create Signature context command and specify the ports of the subsystem.



Finally, you can create the subsystem and its signature in the model from within the Data Dictionary Manager by using the Synchronize System Signature context command. You can also use the `t1SyncSystemSignature` API command.

For further information, refer to

- *Basics on Centrally Specifying Function Subsystem Signatures* ([TargetLink Customization and Optimization Guide](#))
- *How to Specify Function Block Data from Within the Data Dictionary* ([TargetLink Customization and Optimization Guide](#))
- *How to Specify a Subsystem's Signature from within the Data Dictionary Manager* ([TargetLink Customization and Optimization Guide](#))

## Code Generation Core Functionality

### Where to go from here

Information in this section

<i>MISRA-C Compliance</i>	165
<i>Improved Code Efficiency</i>	166

## MISRA-C Compliance

Several improvements were made to TargetLink's Fixed-Point Library to comply with MISRA-C. The improvements include:

- The redundant initializations for the accumulate register were removed from FIR filter macros.
- At many locations in the code, the numeric constants for the data type limits were replaced by global macros (e.g., `INT32MIN`).
- Constants functioning as call parameters were given suffixes or were cast to the expected type if necessary.
- Numeric constants functioning as initial values were given suffixes if necessary.

Further improvements to comply with MISRA-C:

- For macro access functions, the following placeholders for macro arguments are now enclosed by brackets:
  - `_var`
  - `_value`
- For Logical Operator blocks, TargetLink now generates the logical expression `!=` instead of the arithmetic expression `^` to consistently separate logical and arithmetic expressions.
- For relational and logical operations, TargetLink now allows you to avoid assigning the result directly to a non-Boolean variable. This behavior can be controlled via the new `AssignmentOfConditions` Code Generator option.

By default, Relational Operator and Logical Operator blocks now lead directly to `Output = <Condition>;` assignments in production code (independent of optimizations), while

```
if (<Condition>) {
    Output = 1; /* or 0 */
} else {
    Output = 0; /* or 1, respectively */
}
```

is only optimized to an assignment statement if `Output` is of Boolean type (and `Optimization` is enabled).

- For RDI macro definitions, TargetLink now places brackets around the initial values that receive a cast.

## Improved Code Efficiency

---

<b>Loops for nonscalar signals</b>	<p>TargetLink's loop code generation was improved. The improvements include:</p> <ul style="list-style-type: none"><li>■ More block code pattern that lead to element-wise vector assignments or computations in certain configurations now perform these assignments and computations in loops. This applies especially to MinMax, Product, Sum, and Custom Code blocks.</li><li>■ Generating <code>for</code> loops for matrix code, including Stateflow matrix code</li><li>■ Better loop merging</li></ul>
<b>Elimination of unused definitions</b>	<p>TargetLink can now remove unused previous definitions from conditional control flows. This applies primarily to Stateflow outputs driving a Merge block.</p>
<b>Optimization of vectors and vector slices</b>	<p>TargetLink's optimization of vectors and vector slices whose dimensions are smaller than the <code>LoopUnrollThreshold</code> has changed: they are now initially optimized as if their dimensions were equal to or larger than the <code>LoopUnrollThreshold</code>. This allows for vector-specific optimizations in addition to element-wise optimizations.</p>
<b>Implicitly created auxiliary variables</b>	<p>TargetLink can now create implicit auxiliary variables of the vector kind, if this leads to more advantageous block code patterns for subsequent blocks.</p>
<b>Constant indices folding</b>	<p>TargetLink now supports constant folding in indices generated from Assignment and Selector blocks. This improves initial code and does not depend on code optimization.</p> <ul style="list-style-type: none"><li>■ TargetLink 3.5: <code>Sa2_Assignment[1 - 1] =...</code></li><li>■ TargetLink 4.0: <code>Sa2_Assignment[0] =...</code></li></ul>

---

**Substitution of vector variables**

TargetLink now more frequently substitutes vector variables which cannot be eliminated with scalar variables if possible by the code's semantics. This reduces RAM/stack consumption of the generated code:

≤ TargetLink 3.5	TargetLink 4.0
<pre> Int16 vec[..]; loop() {     if (...) {         vec[i] = ...     } else {         vec[i] = ...     }     ... = vec[i]     ... = vec[i] } Int16 vec[..]; loop() {     fnc(&amp;vec[i]);     ... = vec[i]; }                 </pre>	<pre> Int16 scalar; loop() {     if (...) {         scalar = ...     } else {         scalar = ...     }     ... = scalar     ... = scalar } Int16 scalar; loop() {     fnc(&amp;scalar);     ... = scalar; }                 </pre>

## Data Dictionary and Data Management

**Where to go from here**

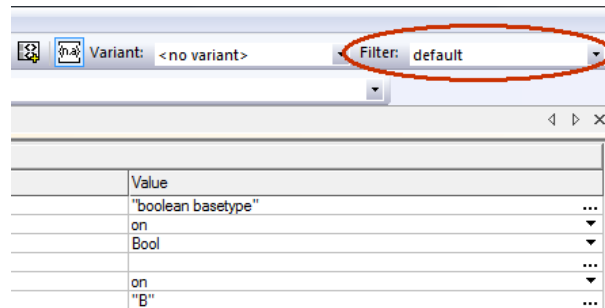
Information in this section

<i>Improvements to the Data Dictionary</i>	168
<i>New DD MATLAB API Commands</i>	170

## Improvements to the Data Dictionary

### Hiding specific objects and properties using filter rule sets

You can now hide specific DD objects and properties in the DD Manager using XML-based filter rule sets. A filter rule specifies whether an object or property defined in the Data Model is visible or invisible. Filter rule sets allow you to customize views for different team members. You can select a filter rule set in the Filter list of the TargetLink Data Dictionary Manager or via the MATLAB API.

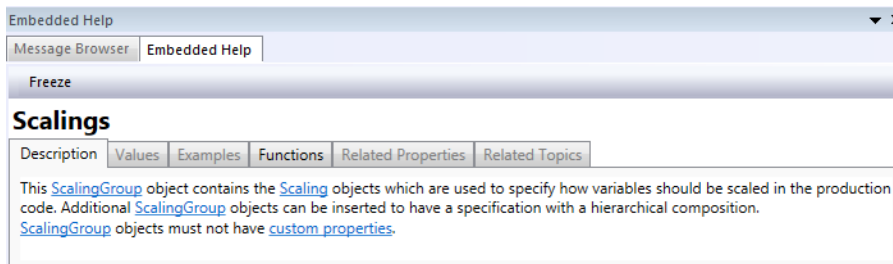


For further information on how to generate filter rule sets, refer to *Basics on Filter Rule Sets for the Data Model* ([TargetLink Data Dictionary Basic Concepts Guide](#)) and *How to Create Filter Rule Sets* ([TargetLink Data Dictionary Basic Concepts Guide](#)).

Examples of filter rule sets can be downloaded from the TargetLink Product Support Center: <http://www.dspace.com/tlpsc>.

### Embedded help for objects and properties

The TargetLink Data Dictionary Manager now provides the Embedded Help pane for DD objects and properties. It provides detailed descriptions on selected objects or properties. By default, the pane is activated. From the Help menu, click Show Embedded Help to deactivate or reactivate it.





For further information, refer to *How to Get Help on DD Objects and Properties* ([📖 TargetLink Data Dictionary Basic Concepts Guide](#)) and *Show Embedded Help* ([📖 TargetLink Data Dictionary Manager Reference](#)).

---

### Improved merge function for objects and properties

A further option was added to the merge functions of the Data Dictionary Manager. The new **Merge and Replace** context command now supports replacing objects and properties. You can

- **Merge <left/right> without overwrite:** The DD objects are merged without overwriting child objects or property values.
- **Merge <left/right>, and overwrite properties:** The DD objects are merged without overwriting DD child objects. Property values are overwritten.
- **Merge <left/right>, and overwrite objects:** The DD objects are merged. Child objects and their property values are overwritten. Values and child objects existing in both DD objects are overwritten.
- **New option: Replace <left/right>:** The DD objects are replaced. The source objects are copied to the target object, replacing the target object.

This context command is also available for the DD Comparison Pane.

For further information, refer to *How to Merge and Replace DD Objects in DD Workspaces* ([📖 TargetLink Data Dictionary Basic Concepts Guide](#)) and *How to Merge and Replace DD Objects in the DD Comparison Pane* ([📖 TargetLink Data Dictionary Basic Concepts Guide](#)).

---

### Extended Recent Files list

The Recent Files list now also shows DD project files opened from within a model or via MATLAB API. Previously, only DD project files opened in the Data Dictionary Manager were shown.

---

### Complete message texts

The messages in the Data Dictionary Manager's **Message Browser** and in the **Custom Output View** are now displayed completely. Previously, only the tooltip of the message contained the complete text.

---

### More robust XML import

The XML import of the Data Dictionary is now more robust to invalid XML files. This is helpful to make sure that XML files exported from previous versions of the Data Dictionary can be imported into the Data Dictionary of TargetLink 4.0.

**Related topics**

## Basics

- *Basics on Filter Rule Sets for the Data Model* ([📖 TargetLink Data Dictionary Basic Concepts Guide](#))

## HowTos

- *How to Create Filter Rule Sets* ([📖 TargetLink Data Dictionary Basic Concepts Guide](#))
- *How to Merge and Replace DD Objects in DD Workspaces* ([📖 TargetLink Data Dictionary Basic Concepts Guide](#))
- *How to Merge and Replace DD Objects in the DD Comparison Pane* ([📖 TargetLink Data Dictionary Basic Concepts Guide](#))

## New DD MATLAB API Commands

TargetLink provides several new DD MATLAB API functions that are listed below. For detailed information, refer to the [📖 TargetLink Data Dictionary MATLAB API Reference](#).

**CountItems**

```
[numOfObjects,numOfProperties] = dsdd('CountItems',<objectIdentifier>);
```

Counts the number of objects and properties in a subtree.

**CreateFilterRuleSet**

```
bSuccess = dsdd('CreateFilterRuleSet',<filterRuleSet>);
```

Creates a filter rule set.

**DeleteFilterRuleSet**

```
bSuccess = dsdd('DeleteFilterRuleSet',<filterRuleSet>);
```

Deletes a filter rule set.

**DumpDataModelPaths**

```
bSuccess = dsdd('DumpDataModelPaths',<file>);
```

Writes all data model paths to a file.

**GetCurrentFilterRuleSet**

```
filterRuleSet = dsdd('GetCurrentFilterRuleSet');
```

Gets a current filter rule set.

**GetDataModelPath**

```
dataModelPath = dsdd('GetDataModelPath',<objectIdentifier>[,<propertyName>]);
```

Gets the data model path of the object or property.

**GetDataModelPaths**

```
dataModelPaths = dsdd('GetDataModelPath',<objectKind>);
```

Gets all data model paths for an object kind.

**GetDataModelTag**

```
dataModelTag = dsdd('GetDataModelTag',<DataModelPath>);
```

Gets the data model tag.

**GetDefaultFilterRule**

```
[bVisible,bValidPath] = dsdd('GetDefaultFilterRule',<DataModelPath>);
```

	Gets the default filter rule of a data model item.
<b>GetFilterRule</b>	<pre>[bVisible,bValidPath] = dsdd('GetFilterRule',&lt;DataModelPath&gt;);</pre> Gets the filter rule of a data model item.
<b>GetFilterRuleChecksum</b>	<pre>checksum = dsdd('GetFilterRuleChecksum');</pre> Gets the data model filter rule checksum.
<b>GetFilterRuleSets</b>	<pre>filterRuleSets = dsdd('GetFilterRuleSets');</pre> Gets the list of filter rule sets.
<b>GetNumOfFilterRuleSets</b>	<pre>numOfFilterRuleSets = dsdd('GetNumOfFilterRuleSets');</pre> Gets the number of filter rule sets.
<b>GetPropertyTable</b>	<pre>propertyTable = dsdd('GetPropertyTable',&lt;objectIdentifier&gt;);</pre> Gets a table with property names and property values.
<b>GetUnsetPropertyNames</b>	<pre>propertyNames = dsdd('GetUnsetPropertyNames',&lt;objectIdentifier&gt;);</pre> Returns the names of properties which are unset.
<b>IsCustomProperty</b>	<pre>bIsCustomProperty = dsdd('IsCustomProperty',&lt;objectIdentifier&gt;,&lt;propertyName&gt;);</pre> Checks if a property is a custom property.
<b>IsVisible</b>	<pre>[bVisible] = dsdd('IsVisible',&lt;objectIdentifier&gt;[,&lt;propertyName&gt;];</pre> Checks if object or property is visible according to the current filter rule set.
<b>ReadFilterRuleSet</b>	<pre>bSuccess = dsdd('ReadFilterRuleSet',attributeName1,attributeValue1,...);</pre> Reads a filter rule set XML file.
<b>ReloadFilterRuleSets</b>	<pre>bSuccess = dsdd('ReloadFilterRuleSets');</pre> Re-reads filter rule set XML files.
<b>RemoveVariants</b>	<pre>errorCode = dsdd('RemoveVariants',&lt;objectIdentifier&gt;[,&lt;propertyName&gt;]);</pre> Removes variants with ID != 0.
<b>Replace</b>	<pre>[hDDObject,errorCode] = dsdd('Replace',[,attributeName1,attributeValue1,...]);</pre> Replaces an object by the copy of another object.
<b>ResetFilterRuleSet</b>	<pre>bSuccess = dsdd('ResetFilterRuleSet',&lt;filterRuleSet&gt;);</pre> Resets a filter rule set.
<b>SetCurrentFilterRuleSet</b>	<pre>bSuccess = dsdd('SetCurrentFilterRuleSet',&lt;filterRuleSet&gt;);</pre>

Sets the current filter rule set.

**SetFilterRule**

```
bValidPath = dsdd('SetFilterRule', <DataModelPath>, <bVisible>);
```

Sets a filter rule of a data model item.

**SetFilterRuleByDataModelTag**

```
nRulesSet = dsdd('SetFilterRuleByDataModelTag', <DataModelTag>, <bVisible>);
```

Sets a filter rule for all items with a specified data model tag.

**WriteFilterRuleSet**

```
bSuccess = dsdd('WriteFilterRuleSet', attributeName, attributeValue1, ...);
```

Writes a filter rule set to an XML file.

## AUTOSAR

**Where to go from here**

Information in this section

<i>Supported AUTOSAR Releases</i>	172
<i>New AUTOSAR Features</i>	173

## Supported AUTOSAR Releases

**Supported AUTOSAR Releases**

The following AUTOSAR Releases are supported:

AUTOSAR Release	Revision
4.1	4.1.3 <sup>1)</sup>
	4.1.2 <sup>1)</sup>
	4.1.1
4.0	4.0.3
	4.0.2
3.2	3.2.3 <sup>1)</sup>
	3.2.2
	3.2.1

AUTOSAR Release	Revision
3.1	3.1.5
	3.1.4
	3.1.2
	3.1.0
	3.0.7
3.0	3.0.6
	3.0.4
	3.0.2
2.1	2.1.4

<sup>1)</sup> New in TargetLink 4.0

### Specifying the AUTOSAR release in the TargetLink Data Dictionary

TargetLink allows you to generate AUTOSAR-compliant code for both AUTOSAR Releases 2.x/3.x and 4.x.

You can specify which AUTOSAR Release to use in the DD /Pool/Autosar/Config object of the TargetLink Data Dictionary.

For information on generating AUTOSAR-compliant code, refer to *Generating AUTOSAR-Compliant Code* ([📖 TargetLink AUTOSAR Modeling Guide](#)).

### Creating AUTOSAR-compliant DD workspaces with system templates

You can now create AUTOSAR-compliant DD workspaces using new system templates for both AUTOSAR Releases 3.x and 4.x. In the TargetLink Data Dictionary Manager, click File - New - Create New DD Workspace and select

- dsdd\_master\_autosar3.dd [System] for AUTOSAR 2.x and 3.x
- dsdd\_master\_autosar4.dd [System] for AUTOSAR 4.x

For further information, refer to *How to Create DD Workspaces* ([📖 TargetLink Data Dictionary Basic Concepts Guide](#)).

## New AUTOSAR Features

### Matrices at AUTOSAR interfaces

TargetLink supports 2-D matrices for all AUTOSAR-related signals with the exception of inter-runnable variables.

### CompuMethods of category IDENTICAL

TargetLink's AUTOSAR import now supports CompuMethods of category IDENTICAL.

<b>Port initialization</b>	TargetLink's AUTOSAR import now supports initialization values which are specified locally at AUTOSAR ports.
<b>Global constants</b>	TargetLink's AUTOSAR import now supports global constants with references to other global constants.
<b>PerInstanceCalPrm</b>	The creation of PerInstanceCalPrm objects in the TargetLink Data Dictionary Manager was simplified.
<b>AUTOSAR memory mapping</b>	You can now use the \$(Component) name macro within the values of the DeclarationStatements property of VariableClass objects to simplify the use of AUTOSAR memory mapping.
<b>Model-linked code view</b>	TargetLink now supports model-linked code view in AUTOSAR code generation mode to provide improved traceability between models and generated code. For details, refer to <i>Tracing Objects between Model and Code (Model-Linked Code View)</i> ( <a href="#">□</a> <i>TargetLink Preparation and Simulation Guide</i> ).
<b>Support of IncludedDataTypeSets</b>	TargetLink now supports IncludedDataTypeSets as described by the AUTOSAR 4.x standard.
<b>Import/export of NetworkRepresentations</b>	TargetLink now imports/exports NetworkRepresentations elements.
<b>Support of SwImplPolicyEnum</b>	TargetLink now provides the following property of DataElement objects:

Property	Value
ImplementationPolicy	<ul style="list-style-type: none"> <li>■ standard - For non-queued sender receiver communication</li> <li>■ queued - For queued sender receiver communication</li> <li>■ measurementPoint - The data element is used for measurement purposes only Only used for AUTOSAR file import/export</li> </ul>



The ImplementationPolicy property replaces the IsQueued property. Refer to *Replaced IsQueued Property* on page 207 for compatibility considerations.

## Test Support

### Where to go from here

Information in this section

<i>Improved Online Parameter Modification</i>	175
<i>Changes in the Target Simulation Modules</i>	175

## Improved Online Parameter Modification

### Automatic update to changed MIL values

Before starting a simulation, you can now have TargetLink update parameter values in a SIL/PIL simulation application automatically. In detail, variable values are then automatically overwritten by their MIL value counterparts if the values' differences have changed and exceed a user-defined tolerance level. In addition, the online parameter update for variables in Stateflow is also supported.

#### Related documentation

- *Basics on Modifying Parameter Values for Simulation*  
(📖 *TargetLink Preparation and Simulation Guide*)
- *How to Provide Automatic Parameter Updates via a Hook Function*  
(📖 *TargetLink Preparation and Simulation Guide*)

## Changes in the Target Simulation Modules

### New and discontinued compiler versions

The following table shows the compiler versions that are now supported by TargetLink 4.0, refer to the **New** and **No changes** columns. Compiler versions that are no longer supported are listed in the **Discontinued** column.

Target	Compiler	New	No changes	Discontinued
C16x	TASKING	—	8.6, 8.7	—
HCS12	Cosmic	—	4.8	4.7
	Metrowerk	—	5.1	3.1
M32R	Gaio	—	11, 9	—
	Renesas	—	5.1	—
MC56F83	Metrowerk	—	8.3	—

Target	Compiler	New	No changes	Discontinued
MPC55xx	Diab	—	5.9	5.7
	GreenHill	2013	—	2012
	GNU	—	4.1	—
	Metrowerk	—	2.8	—
MPC55xxVLE	Diab	—	5.9	—
	GreenHill	2013	—	2012
	Metrowerk	—	2.8	—
MPC560xVLE	Diab	—	5.9	—
	GreenHill	2013	2012	5.2
MPC5xx	Diab	—	—	5.7
	GreenHill	—	—	5.1
S12X	Cosmic	—	4.8	—
	Metrowerk	—	5.1	—
SH2	Renesas	—	9.3, 9.4	—
SH2A-FPU	Renesas	—	9.4	—
TriCore17xx	TASKING	4.3	3.2	4.2
TriCore1796	GNU	—	3.4	—
V850	GreenHill	2013	—	2012
	NEC	—	3.4	—
XC22xx	TASKING	—	3.0	—

For detailed information on the evaluation boards supported by TargetLink, refer to [TargetLink Evaluation Board Hardware Reference](#).



The MPC5xx target is no longer supported by TargetLink but still distributed by dSPACE.

For further PIL support combinations that are part of a valid Software Maintenance Service (SMS) contract, refer to dSPACE's TargetLink PIL Support website at the TargetLink Product Support Center.



# Code Generator Options

## New Code Generator Options

The following new Code Generator options are available with TargetLink 4.0.

Description	Explanation	Default
<b>AssignmentOfConditions</b>		
<p>Controls the code pattern for Relational Operator and Logical Operator blocks and related optimizations.</p>	<p>For Relational Operator and Logical Operator blocks, TargetLink lets you decide whether you want code of the form</p> <pre>if (&lt;Condition&gt;) {     Output = 1; } else {     Output = 0; }</pre> <p>or of the form</p> <pre>Output = &lt;Condition&gt;;</pre> <p>In addition, you can activate a control flow optimization that transforms the former code pattern into the latter, e.g., for code stemming from a Stateflow chart or a Switch block with one 0 and one 1 data input.</p> <p>The option values have the following effect:</p> <ul style="list-style-type: none"> <li>■ 0 - None: Always generate the control flow pattern for Relational Operator and Logical Operator blocks and do not perform control flow optimization.</li> <li>■ 1 - RelationAndLogicBlocks: Generate the assignment pattern for Relational Operator and Logical Operator blocks where possible but do not perform control flow optimization.</li> <li>■ 2 - AllBooleanOutputs: Generate code as for the 'RelationAndLogicBlocks' setting and perform the control flow optimization if the output variable is of Boolean type.</li> <li>■ 3 - AllOutputs: Generate code as for setting 'RelationAndLogicBlocks' and perform the control flow optimization if the output variable is of any numerical type.</li> </ul> <p>The last setting might unexpectedly lead to code that violates MISRA rules and performs arithmetic or bitwise operations with operands that are logic or relation operations.</p>	2

Description	Explanation	Default
<b>InsertComputeThroughOverflowComments</b>		
Introduce additional code comments to mark casts introduced for Compute Through Overflow operations.	To produce more efficient code, TargetLink allows controlled overflows in the calculation of additions/subtractions (Compute Through Overflow). When this option is enabled, comments are introduced in the generated code at casts that introduce these controlled overflows. <ul style="list-style-type: none"> <li>■ 0 - off</li> <li>■ 1 - on</li> </ul>	off
<b>Allow64BitMultiplicationsForArbitraryScaled16BitOperands</b>		
Controls the code pattern of product operation with 16-bit operands.	A product operation with 16-bit operands and an arbitrary scaling usually returns 64-bit code in order to generate code with the highest possible precision for the rescaling operations. If 64-bit code is to be avoided, this option allows to suppress this behavior for scale factors $\leq 1$ . This might cause a great loss of accuracy. <ul style="list-style-type: none"> <li>■ 0 - off</li> <li>■ 1 - on</li> </ul>	on
<b>UtilizeExplicitDependenciesForScalingInvariantSystems</b>		
Considers optionally specified information on which inport(s) a scaling invariant output depends on.	When this option is set to on, the optionally in the scaling propagation function specified information, specifying the inport(s) a scaling invariant output depends on, is taken into account for the code generation. Otherwise, a worst case assumption (i.e., each output is dependent on every inport of the scaling invariant system) is made for a scaling invariant system. Specifying the dependencies is necessary for the modeling of feedback loops with scaling invariant systems. <ul style="list-style-type: none"> <li>■ 0 - off</li> <li>■ 1 - on</li> </ul>	off

For reference information on all Code Generator options, refer to *Code Generator Options* ([📖 TargetLink Block and Object Reference](#)).

**Migration aspects of Code Generator options**

Migration aspects include:

- Obsolete Code Generator options
- Changed Code Generator options
- Recommended compatibility options for new Code Generator options to ensure the best possible downward compatibility in the generated production code

Refer to *Migration Aspects Regarding Code Generator Options* on page 200 for more information.

# Tool Chain Integration

## Improved Windows Compliance

---

### Installation files in user-defined locations

This TargetLink version can search for the following customization files at arbitrary locations, i.e., outside your TargetLink installation:

- DD template files
- DD Data Model filter rules
- DD menu extensions
- A2L style sheets

#### Related documentation

- *How to Define TargetLink's Search Path for DD-Related Customization Files* ([📖 TargetLink Customization and Optimization Guide](#))

---

### Import/export of TargetLink Preferences

You can now import and export the TargetLink preferences settings also via command line.

#### Related documentation

- *Basics on Using the TargetLink Preferences Editor* ([📖 TargetLink Customization and Optimization Guide](#))
- *How to Import and Export TargetLink Preferences via Command Line* ([📖 TargetLink Customization and Optimization Guide](#))

---

### Cloning TSMs

You can clone already installed TSMs and store them to the TSM Extension Packages folder (which can be specified in the TargetLink Preferences Editor). They are then also listed in the Target Simulation Module > Preselection dialog of the TargetLink Preferences Editor. Additionally, you can remove TSMs that do not belong to the TargetLink installation, for example, TSMs that were created by cloning an installed TSM or added as additional TSM packages.

#### Related documentation

- *How to Clone Target/Compiler Combinations to Outside the TargetLink Installation* ([📖 TargetLink Customization and Optimization Guide](#))

## Other

### General Enhancements and Changes

---

#### Improved documentation generation

The following new features are available for TargetLink's documentation generation:

**Support for Japanese characters in PDF generation** TargetLink now also supports Japanese characters in the PDF document generation process. By default, TargetLink automatically detects the installed language of Microsoft® Windows® and then sets the appropriate character set and font. However, you can manually set the `CharacterSet` property in the `Config/General` object of the DD object tree. In addition, you have to set the font to be used in the generated PDF documentation in the script that controls the PDF documentation generation (default: `tldoc_pdf`).

**New table of contents and optional cover page** PDF document generation now provides a table of contents. The depth of the table of contents can be set the user (see `tldoc_pdf` script for details)

Optionally, you can also add a cover page to the generated PDF. To do so, edit the `tldoc_pdf` M script and add a `CoverPage` entry as described in the script.

**Disabling the generation of the function hierarchy** You can now disable the generation of a hierarchical list of functions contained in the generated code. By default, the generation is activated. To disable it, edit the M script that controls the generation of the documentation (for example `tldoc_default`, `tldoc_pdf`, `tldoc_rtf`) and set the `FunctionsHierarchy` entry to 'off'.

**Order of user-inserted chapters in the function list** In former TargetLink versions, user-inserted chapters that were created via the `Autodoc Customization` block were not integrated in the navigation pane of the generated documentation. They were listed at the end of the contents. Now user-inserted chapters are integrated by default. This only applies to HTML-generated documentation.

For further information on the document generation process, refer to *Basics on Customizing the Generated Documentation* ([📖 TargetLink Interoperation and Exchange Guide](#)).

<b>Improved consistency check for indices</b>	<p>TargetLink now performs consistency checks for indices during code generation.</p> <p>For Assignment blocks and Simulink Selector blocks, TargetLink now performs a consistency check for indices.</p> <p>If the block's index mode does not match the signal at the block's index port, an error message occurs.</p>
<b>Creating customization files via GUI</b>	<p>TargetLink provides a tool (GUI) to derive customization files from templates. In addition, you can use this tool to create DD menu extensions and A2L style sheets.</p> <p><b>Related documentation</b></p> <ul style="list-style-type: none"> <li>■ <i>How to Create TargetLink Customization Files</i> (<a href="#">📖 TargetLink Customization and Optimization Guide</a>)</li> </ul>
<b>Obsolete limitations</b>	<p>With TargetLink 4.0, several limitations were removed. For details, refer to <i>Obsolete Limitations</i> on page 214.</p>

## API Commands

### New API Functions

<b>t1ExtractSubsystem</b>	<p>The <b>t1_extract_subsystem</b>, API function (TargetLink ≤3.5) was replaced.</p> <p>TargetLink provides the <b>t1ExtractSubsystem</b> (<a href="#">📖 TargetLink API Reference</a>) API function to generate a new independent TargetLink subsystem from a TargetLink subsystem part.</p> <p><b>Related documentation</b></p> <ul style="list-style-type: none"> <li>■ <i>t1ExtractSubsystem</i> (<a href="#">📖 TargetLink API Reference</a>)</li> </ul>
<b>t1SimInterface</b>	<p>The <b>t1_sim_interface</b> API function (TargetLink ≤3.5) was replaced.</p>

TargetLink provides the **tlSimInterface** ([TargetLink API Reference](#)) API function as M interface for the TargetLink simulation engine.

### Related documentation

- *Implementing Online Parameter Modification* ([TargetLink Preparation and Simulation Guide](#))
- *tlSimInterface* ([TargetLink API Reference](#))

### tlSimParameterUpdate

TargetLink provides the **tlSimParameterUpdate** ([TargetLink API Reference](#)) API function to allow automatic online parameter modification.

### Related documentation

- *Implementing Online Parameter Modification* ([TargetLink Preparation and Simulation Guide](#))
- *tlSimParameterUpdate* ([TargetLink API Reference](#))

### tlMoveDDObject

TargetLink provides the **tlMoveDDObject** ([TargetLink API Reference](#)) API function to move or rename Data Dictionary objects and adapt references to this object.

### Related documentation

- *tlMoveDDObject* ([TargetLink API Reference](#))
- *tlFindDDRReferences* ([TargetLink API Reference](#))

### tlSyncSystemSignature

TargetLink provides the **tlSyncSystemSignature** ([TargetLink API Reference](#)) API function to generate a new Simulink system from a Data Dictionary Signature or Block object.

### Related documentation

- *tlSyncSystemSignature* ([TargetLink API Reference](#))

### tlOperationMode

The **tl\_switch\_blockset** API function (TargetLink ≤3.5) was replaced.

TargetLink provides the **tlOperationMode** ([TargetLink API Reference](#)) API function to get or set TargetLink's operation mode (*full-featured* versus *stand-alone*).

### Related documentation

- *tlOperationMode* ([TargetLink API Reference](#))

### tlUpgrade

The **tl\_upgrade** API function (TargetLink ≤3.5) was replaced.

TargetLink provides the **t1Upgrade** API function to upgrade block diagrams containing TargetLink blocks to the current version.

### Related documentation

- *tUpgrade* ([📖 TargetLink API Reference](#))

---

### T1TsmManager

TargetLink provides the **T1TsmManager.exe** command to clone existing TargetLink Simulation Modules.

### Related documentation

- *How to Clone Target/Compiler Combinations to Outside the TargetLink Installation* ([📖 TargetLink Customization and Optimization Guide](#))

# Migrating to TargetLink 4.0 and TargetLink Data Dictionary 4.0

## No upgrade from TargetLink 2.x



Libraries, models, and DD files from TargetLink versions prior to TargetLink 3.1 cannot be upgraded directly. However, you can perform an intermediate upgrade. Migrate older libraries, models, and DD files to TargetLink 3.5. You can then upgrade them to TargetLink 4.0

## Where to go from here

Information in this section

<i>Data Dictionary and Data Management</i>	184
<i>Code Changes</i>	189
<i>Code Generator Options</i>	200
<i>Changes With Access Functions</i>	201
<i>AUTOSAR-Related Migration Aspects</i>	204
<i>Other</i>	207
<i>API Commands</i>	212
<i>Obsolete</i>	213
<i>Messages</i>	215
<i>Stateflow-Related Changes</i>	216

## Data Dictionary and Data Management

## Where to go from here

Information in this section

<i>Migrating to TargetLink 4.0</i>	185
<i>How to Upgrade a Data Dictionary With Included DD Files</i>	186

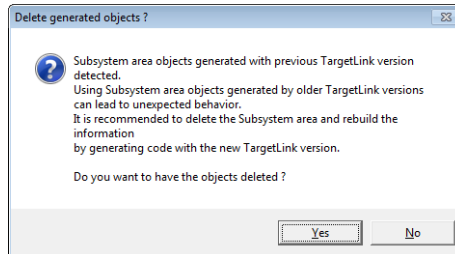
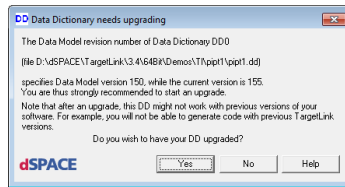


## Migrating to TargetLink 4.0



TargetLink 4.0 automatically upgrades models, libraries, or Data Dictionaries created with TargetLink 3.1 or higher.

The following dialogs can appear during the upgrade process:



**User interaction** In the following cases, user interaction is required:

- *Legacy libraries never prepared for TargetLink* on page 185
- *Migrating from TargetLink 32-bit to TargetLink 64-bit* on page 186
- *DD files with included partial DD files* on page 186
- *Changes with access functions* on page 186

### Legacy libraries never prepared for TargetLink

Libraries created with TargetLink 3.x that were never prepared using the `tl_prepare_system` ([TargetLink API Reference](#)) API function cannot be upgraded automatically by TargetLink 4.0.

#### Solution

1. Open the library in the prior TargetLink version and prepare it for the upgrade by using `tl_prepare_system` ([TargetLink API Reference](#)).
2. Save the library.
3. Open the library with TargetLink 4.0.

### Related documentation

- *How to Make TargetLink User Libraries Upgrade-Capable*  
(📖 *TargetLink Orientation and Overview Guide*)

#### Migrating from TargetLink 32-bit to TargetLink 64-bit

Custom code S-functions built with 32-bit TargetLink versions do not work with 64-bit versions of TargetLink and vice-versa.

**Solution** Initiate a rebuild of all custom code S-functions using the `tlUpgrade('Model', <MyModel>, 'CheckModel', 'FixIssues')` (refer to `tlUpgrade` (📖 *TargetLink API Reference*)) API function.

#### DD files with included partial DD files

To upgrade DD files with included partial DD files, refer to *How to Upgrade a Data Dictionary With Included DD Files* on page 186.

#### Changes with access functions

Access functions of the `ADDRESS_BY_PARAMETER` kind have changed. User interaction might be required. For details, refer to *Changes of ADDRESS\_BY\_PARAMETER Access Functions* on page 202.

## How to Upgrade a Data Dictionary With Included DD Files

### Objective

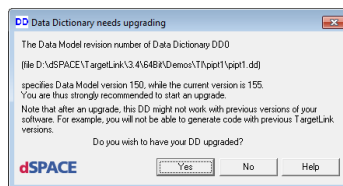
If you open a TargetLink model with an old, non-upgraded Data Dictionary file, you have to upgrade the Data Dictionary file.

### Method

#### To upgrade a Data Dictionary with included DD files

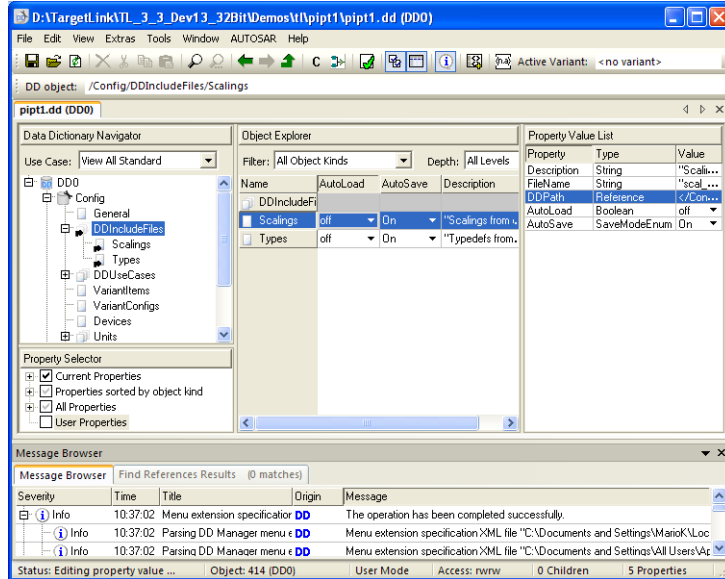
- 1 Open the model and the referenced TargetLink Data Dictionary, or type `dsdd('Open', <DDFile>)` in the MATLAB Command Window.

The Data Dictionary needs upgrading dialog automatically opens if an earlier DD version is involved.



- 2 Select No in the upgrade dialog.

- Under /Config/DDIncludeFiles, set the AutoLoad and AutoSave properties for each included DD file as shown in the following screenshot.

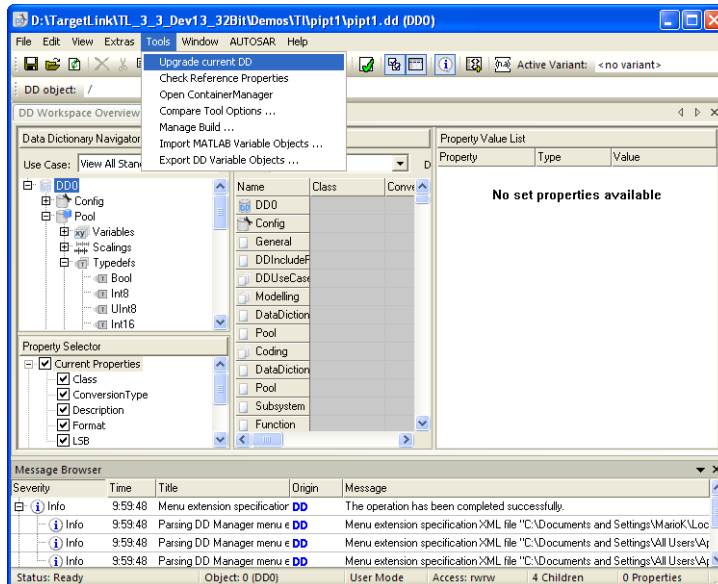


This ensures that after the Data Dictionary and the included DD files have been upgraded, the upgraded included DD files are saved when the Data Dictionary is saved. You can set these properties for a large number of included DD files via the Object Explorer.



You can also use the Point of Inclusion dialog to set the included DD file properties.

- Start the DD upgrade (with the included DD files) via Tools - Upgrade current DD in the DD Manager, or enter `dsdd('Upgrade')` in the MATLAB Command Window.



- Save the Data Dictionary (with write permission to the relevant DD file). This completes the upgrade of the DD file and the included partial DD files.

## Result

When you open the DD file again, the upgrade dialog will not open because the DD file and the included partial DD files are up-to-date. Once the files have been properly upgraded, you might want to restore the old settings for the included DD files.

# Code Changes

## Code Changes

**ADDRESS\_BY\_PARAMETER access functions**      The type of the `_var` macro argument changed from `scalar` type to pointer to scalar type for *vector variables*. For details, refer to *Changes of ADDRESS\_BY\_PARAMETER Access Functions* on page 202.

**Display of signal widths in block comments**      Block comments no longer provide information on the signal indices.

≤ TargetLink 3.5	TargetLink 4.0
<pre>for (Aux_S32 = 0; Aux_S32 &lt; 5; Aux_S32++) {     /* Gain: Subsystem/Gain [0..4] */     Sal_Gain[Aux_S32] = Sal_InPort[Aux_S32] * 3;      /* Gain: Subsystem/Gain [5..9] */     Sal_Gain[Aux_S32 + 5] = Sal_InPort[Aux_S32 + 5] * 5; }</pre>	<pre>for (Aux_S32 = 0; Aux_S32 &lt; 5; Aux_S32++) {     /* Gain: Subsystem/Gain */     Sal_Gain[Aux_S32] = Sal_InPort[Aux_S32] * 3;     Sal_Gain[Aux_S32 + 5] = Sal_InPort[Aux_S32 + 5] * 5; }</pre>

**Block comments in if-else branches**      Block comments are now consistently placed at each statement:

≤ TargetLink 3.5	TargetLink 4.0
<pre>/* Switch: Subsystem/Switch1 */ if (c == 0) {     ... &lt;code&gt; ... } else {     ... &lt;code&gt; ... }</pre>	<pre>/* Switch: Subsystem/Switch1 */ if (c == 0) {     /* Switch: Subsystem/Switch1 */     ... &lt;code&gt; ... } else {     /* Switch: Subsystem/Switch1 */     ... &lt;code&gt; ... }</pre>

**Vector versus matrix variables**      If the Interpret as 1D flag is not set, TargetLink treats block variables (e.g., at the Constant block) as follows (in accordance to Simulink):

≤ TargetLink 3.5	TargetLink 4.0
Creates a <i>vector</i> variable.	Creates a <i>matrix</i> variable.

## Muxed signals at Simulink Inports/Outports

If muxed signals are fed to an unenhanced in-/outport block of an atomic subsystem, TargetLink implements the whole block either as one vectorized function parameter or as one vectorized global variable:

≤ TargetLink 3.5	TargetLink 4.0
<p>E.g., one function parameter for each muxed signal, or in some cases a struct:</p> <pre>Void Sa2_Inner(Int16 Sa2_InPort[2], Int16 Sa2_InPort_a);</pre> <p>— or —</p> <pre>struct BS_IS_Sal_BusInport {     Int16 Sal_Signal1[2];     Int16 Sal_Signal2; } [...]</pre> <pre>Void Sa2_Inner(struct BS_IS_Sal_BusInport * Sa2_InPort);</pre>	<p>E.g., one function parameter for the whole port:</p> <pre>Void Sa2_Inner(Int16 Sa2_InPort[3]);</pre>

## Property inheritance and mapping struct components to variables

If a block inherits its data types via Inherit properties from a bus port block whose bus elements are mapped on one DD structure, and TargetLink cannot propagate the struct type, then TargetLink generates a separate variable for each DD structure component, for example:

≤ TargetLink 3.5	TargetLink 4.0
<pre>static Int16 X_Sal_Unit_Delay[2] = {     0, 0 }; [...]</pre> <pre>Sal_a = X_Sal_Unit_Delay[0]; X_Sal_Unit_Delay[0] = StructVar.a; Sal_b = X_Sal_Unit_Delay[1]; X_Sal_Unit_Delay[1] = StructVar.b;</pre>	<pre>static Int16 X_Sal_Unit_Delay = 0; static Int16 X_Sal_Unit_Delay_a = 0; [...]</pre> <pre>Sal_a = X_Sal_Unit_Delay; X_Sal_Unit_Delay = StructVar.a; Sal_b = X_Sal_Unit_Delay_a; X_Sal_Unit_Delay_a = StructVar.b;</pre>

## Pointer-to-const return type for RTE API functions

For `Rte_IRead`, `Rte_Calprm`, and `Rte_CData` function calls using non-scalar data types, TargetLink uses the Pointer-to-const return type if the `/Pool/Autosar/Config/UseRtePointerToConstForNonScalarReturnValues` DD property is set to `on`.

≤ TargetLink 3.5	TargetLink 4.0
<pre>StructWSubStructsType * DE_StructWSubStructs; DE_StructWSubStructs = Rte_IRead_Swc_Run_ReceiverPort_DE_StructWSubStructs();</pre>	<pre>const StructWSubStructsType * DE_StructWSubStructs; DE_StructWSubStructs = Rte_IRead_Swc_Run_ReceiverPort_DE_StructWSubStructs();</pre>

**Pointer-to-const function signatures for RTE API functions**

IN function parameters that are read-only, and of non-scalar data types, provide an additional qualifier (const), if the /Pool/Autosar/Config/UseRtePointerToConstForInArguments DD property is set to on.

≤ TargetLink 3.5	TargetLink 4.0
Rte_<Fcn>( ..., <Vector Address>, ... );	Rte_<Fcn>( ..., (const T*) <Vector Address>, ... );

**Pointers to AUTOSAR return variables**

A new name template is used for pointers to AUTOSAR pointer return variables in order to improve readability and to avoid global name collisions.

≤ TargetLink 3.5	TargetLink 4.0
\$(DataElement)\$R	Local pointer: p_\$(DataElement)\$R  Global pointer: p_I_\$(DataElement)\$R

**Selector Block**

If TargetLink generates an implicit variable for the input, its name has now the suffix \_In, which is more intuitive.

≤ TargetLink 3.5	TargetLink 4.0
Int16 Sa1_Selector_a[20];	Int16 Sa1_Selector_In[20];

**Rate Limiter Block**

An implicit variable is generated for the block input only if it is required, i.e., when the input has a different data type or scaling, or when saturation is enabled for the output.

**DisableArbitraryOptimizations Code Generator option**

DisableArbitraryOptimizations is renamed in Allow64BitMultiplicationsForArbitraryScaled16BitOperands, and the default setting was changed from off to on. In consequence, TargetLink generates more 64-bit operations, but the precision of simulation results improves as rescaling becomes more accurate.

**Direct LUT block**

For Direct LUT block pattern with `Slices < LoopUnrollThreshold`, this TargetLink version applies the following code generation approach:

- Fewer implicit variables for the indices
- Changed sequence of code lines for improved efficiency
- More code comments for improved traceability

≤ TargetLink 3.5	TargetLink 4.0
<pre> Aux_U8 = C_U8SATTI16_SATu(Sal_IndexID_inport[0], 10); Aux_U8_a = C_U8SATTI16_SATu(Sal_IndexID_inport[1], 10); Aux_U8_b = C_U8SATTI16_SATu(Sal_IndexID_inport[2], 10);  /* TargetLink output: RootSystem/OutPort2 # combined # RootSystem/Direct Look-Up Table (n-D) Ext */ Sal_OutPort2[0] = Sal_TableData_inport[Aux_U8]; Sal_OutPort2[1] = Sal_TableData_inport[Aux_U8_a]; Sal_OutPort2[2] = Sal_TableData_inport[Aux_U8_b];         </pre>	<pre> Aux_U8 = C_U8SATTI16_SATu(Sal_IndexID_inport[0], 10); /* RootSystem/Direct Look-Up Table (n-D) Ext # combined # TargetLink output: RootSystem/OutPort2 */ Sal_OutPort2[0] = Sal_TableData_inport[Aux_U8];  Aux_U8 = C_U8SATTI16_SATu(Sal_IndexID_inport[1], 10); /* RootSystem/Direct Look-Up Table (n-D) Ext # combined # TargetLink output: RootSystem/OutPort2 */ Sal_OutPort2[1] = Sal_TableData_inport[Aux_U8];  Aux_U8 = C_U8SATTI16_SATu(Sal_IndexID_inport[2], 10); /* RootSystem/Direct Look-Up Table (n-D) Ext # combined # TargetLink output: RootSystem/OutPort2 */ Sal_OutPort2[2] = Sal_TableData_inport[Aux_U8];         </pre>

**Improved transparency of implicit AUX variables**

For functions, implicit variables are initially always created at function level. Activating the Code Generator option

`ReduceScopeOfVariablesOnlyDownToFunctionLevel` now affects all variables. If the options

`ReduceScopeOfVariablesOnlyDownToFunctionLevel` and `Optimization` have default values, changes can only be observed for implicit vector variables. This can lead to changes in the number of `Aux_` variables and/or in their names.

≤ TargetLink 3.5	TargetLink 4.0
<pre> MyFunc () {     ...     if (...)     {         Int32 Aux_S32;          Aux_S32 = IN * 42;         MinOut = C_16FITI32_SAT(Aux_S32, ...);     } }         </pre>	<pre> MyFunc () {     Int32 Aux_S32;     ...     if (...)     {         Aux_S32 = IN * 42;         MinOut = C_16FITI32_SAT(Aux_S32, ...);     } }         </pre>



**Actual parameters of function return values in incremental systems**

Suppose the following:

- The output of an incremental system is specified as a function return value, and
- The name of the actual parameter, i.e., the variable appearing in the function call statement, is not explicitly specified, or specified as `<FormalparameterName>$R`, and
- In the code generated for the incremental system, a different variable is used as the return value because the original variable has been deleted due to optimization.

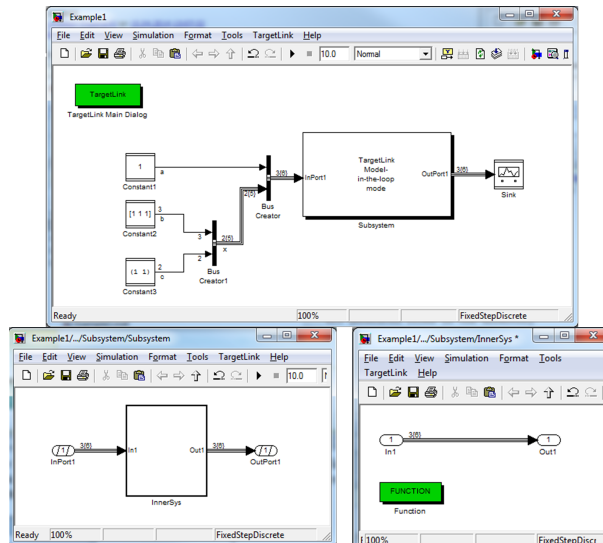
Hence, the actual parameter of the wrapping system is named as follows (independently of whether the system is incremental or not):

≤ TargetLink 3.5	TargetLink 4.0
<FormalparameterName>	<FormalparameterName>_a

**Implicit interface variables for Simulink inports/outports**

For de-enhanced inports and outports of atomic subsystems, the code generated for implicit interface variables (IF variables) differs:

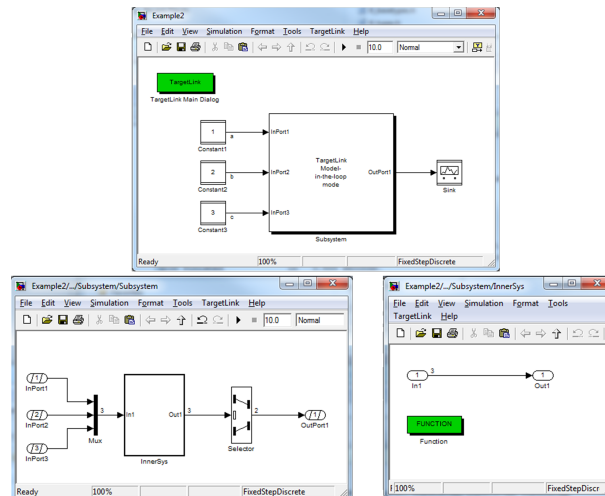
**Bus signals**



If a nested bus signal (a; (b[3];c[2])) is fed to a de-enhanced output, then the IF variable for the output is generated as follows:

≤ TargetLink 3.5	TargetLink 4.0
One IF variable for the whole bus signal: static Int16 IF_Sa2_Out1[6];	An IF variable for each bus element: static Int16 IF_Sa2_Out1; static Int16 IF_Sa2_Out1_a[3]; static Int16 IF_Sa2_Out1_b[2];

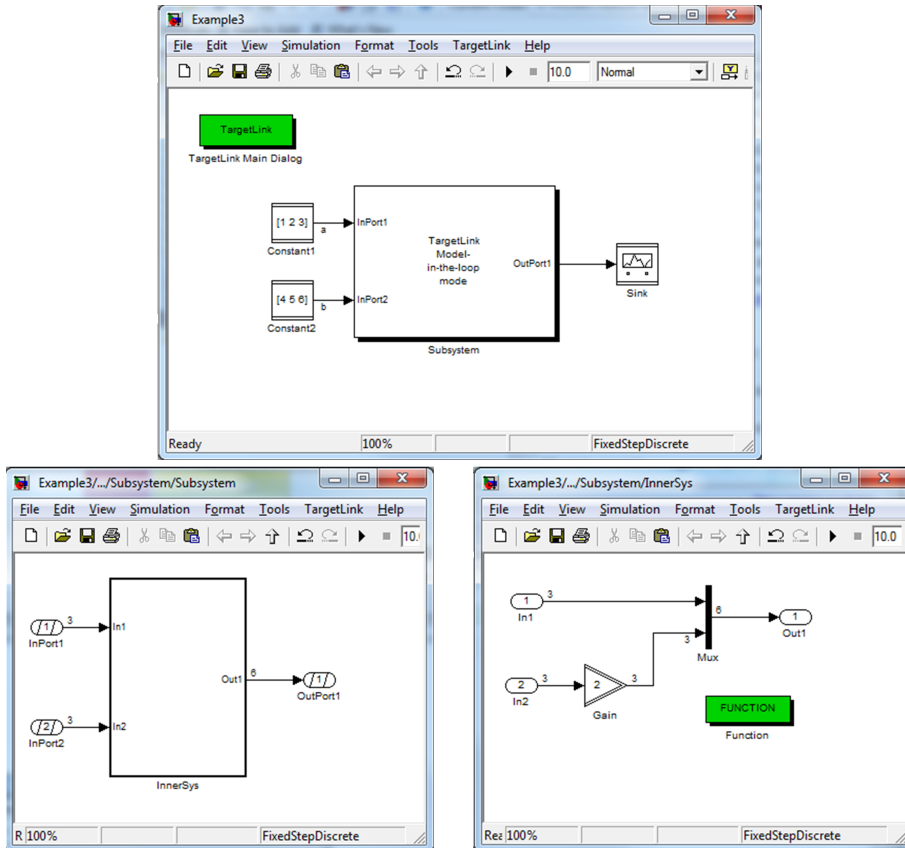
### Muxed signals in combination with Selector block



If muxed scalar signals (a; b; c) are fed to a de-enhanced output, then the IF variable for the output is generated as follows:

≤ TargetLink 3.5	TargetLink 4.0
Two IF scalar variables: static Int16 IF_Sa2_Out1; static Int16 IF_Sa2_Out1_b;	One IF vector variable: static Int16 IF_Sa2_Out1[3];

### Muxed signals arising from different blocks inside an atomic subsystem



If muxed vector signals (a[3]; b[3]) are fed to a de-enhanced output, then the IF variable for the output is generated as follows:

≤ TargetLink 3.5	TargetLink 4.0
One IF vector variable: <pre>static Int16 IF_Sa2_Gain[3]; static Int16 IF_Sa2_Out1[3];</pre>	Three IF scalar variables: <pre>static Int16 IF_Sa2_Gain[3]; static Int16 IF_Sa2_Out1; static Int16 IF_Sa2_Out1_a; static Int16 IF_Sa2_Out1_b;</pre>

## No negative offsets in loop assignments

For assignments within for loops, TargetLink does not generate negative offsets:

≤ TargetLink 3.5	TargetLink 4.0
<pre>for(int Aux__a= 4; Aux__a &lt; 10; Aux__a++) {     Aux[Aux__a - 3] = In[Aux__a] }</pre>	<pre>for(int Aux__a= 1; Aux__a &lt; 7; Aux__a++) {     Aux[Aux__a] = In[Aux__a + 3] }</pre>

## Mitigation of qualifier loss

TargetLink introduces an implicit variable to mitigate qualifier loss if the qualifier is const:

≤ TargetLink 3.5	TargetLink 4.0
<pre>const volatile Int16 MyConstVolArray[3] = {0,0,0}; void Fcn(volatile Int16 * pFormalParam);  Fcn((volatile Int16*) MyConstVolArray); /* Loss of const */</pre>	<pre>const volatile Int16 MyConstVolArray[3] = {0,0,0}; void Fcn(volatile Int16 * pFormalParam);  Int16 MyConstVolArray_Aux[3]; for i = 0:2     MyConstVolArray_Aux[i] = MyConstVolArray[i]; Fcn((volatile Int16 *)MyConstVolArray_Aux);</pre>

When such an implicit variable is introduced, the message A17363 is emitted by the Code Generator to inform you about this possible inconsistent specification and solutions.

These implicit variables can occur in the following contexts:

- Functions of the FIR Filter block
- Subsystems that contain a TargetLink Function block
- Reused functions whose reuse structures were set to const via templates
- RTE calls resulting from model entities placed immediately before subsystems

The following contexts are not analyzed:

- Custom look-up script functions
- Custom code inputs
- Stateflow Extern C Functions

## Bitfield casts

TargetLink now eliminates superfluous unsigned int casts in assignments involving bitfields, such as BitfieldVar = BoolVar; or BitfieldVar = <BoolOperation>;.

<BoolOperation> stands for one of the following>, >=, ==, !=, <=, <, &&, ||, !

≤ TargetLink 3.5	TargetLink 4.0
<pre>GIBFS_b_X_Sb21_Memory_U = (unsigned int) T_SwHDIPushed_b;</pre>	<pre>GIBFS_b_X_Sb21_Memory_U = T_SwHDIPushed_b;</pre>

**Removal of EfficientVectorHandling**

The EfficientVectorHandling Code Generator option was removed from TargetLink 4.0.

TargetLink now always checks the dimensions of vectors and matrices against the LoopUnrollThreshold and generates a for loop for dimensions greater than or equal to the defined threshold. For details, refer to *Migration Aspects Regarding Code Generator Options* on page 200.

Moreover, Stateflow signals and Simulink signals are now treated equally with respect to loop generation. This is especially visible in code generated for Stateflow.

Code changes cannot be ruled out if EfficientVectorHandling was set to off or if the vector signal's dimension was smaller than the LoopUnrollThreshold:

≤ TargetLink 3.5	TargetLink 4.0
<pre>do {   Cal_a[idx1] = Cal_b[idx1];   if (Cal_b[idx1] &lt; 0) {     Cal_a[idx1] = -Cal_a[idx1];   }   idx1++; } while (idx1 &lt; 4);</pre>	<pre>Cal_a[0] = Cal_b[0]; Cal_a[1] = Cal_b[1]; Cal_a[2] = Cal_b[2]; Cal_a[3] = Cal_b[3]; if (Cal_b[0] &lt; 0) {   Cal_a[0] = -Cal_a[0]; } if (Cal_b[1] &lt; 0) {   Cal_a[1] = -Cal_a[1]; } if (Cal_b[2] &lt; 0) {   Cal_a[2] = -Cal_a[2]; } if (Cal_b[3] &lt; 0) {   Cal_a[3] = -Cal_a[3]; }</pre>

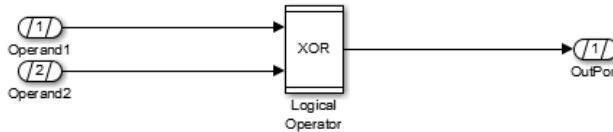
In TargetLink 4.0 for loops are generated according to the principles described in *Basics on Processing Vectors and Matrices* (📖 *TargetLink Preparation and Simulation Guide*):

Dimension < LoopUnrollThreshold	Dimension ≥ LoopUnrollThreshold
<pre>Cal_a[0] = Cal_b[0]; Cal_a[1] = Cal_b[1]; Cal_a[2] = Cal_b[2]; Cal_a[3] = Cal_b[3]; if (Cal_b[0] &lt; 0) {   Cal_a[0] = -Cal_a[0]; } if (Cal_b[1] &lt; 0) {   Cal_a[1] = -Cal_a[1]; } if (Cal_b[2] &lt; 0) {   Cal_a[2] = -Cal_a[2]; } if (Cal_b[3] &lt; 0) {   Cal_a[3] = -Cal_a[3]; }</pre>	<pre>for (Aux_S32 = 0; Aux_S32 &lt; 4; Aux_S32++) {   Cal_a[Aux_S32] = Cal_b[Aux_S32];   if (Cal_b[Aux_S32] &lt; 0) {     Cal_a[Aux_S32] = -Cal_a[Aux_S32];   } }</pre>

**Logical Operator block (exclusive OR)**

The code generated for Logical Operator blocks whose Operator is set to XOR changed.

TargetLink now generates the logical expression != instead of the arithmetic expression ^ to consistently separate logical and arithmetic expressions. This also means improved MISRA-C compliance.



≤ TargetLink 3.5	TargetLink 4.0
<pre>/* Logical: Logical Operator TL_Root/Logical Operator */ block_out = (!Operand1) ^ (!Operand2);</pre>	<pre>/* Logical: TL_Root/Logical Operator */ block_out = (Operand1 != 0) != (Operand2 != 0);</pre>

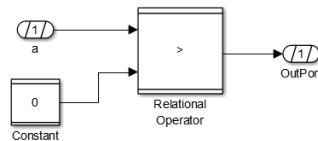
**Rescheduling of state updates**

TargetLink's analysis for state update rescheduling has improved. This can result in a different order of state updates in the code.

**Relations and logical operations**

TargetLink's code pattern and optimization behavior for Logical Operator and Relational Operator blocks changed.

**Example**



**Optimization = OFF**

≤ TargetLink 3.5	TargetLink 4.0
<pre>if (a &gt; 0) {     Sal_Relational_Operator = 1; } else {     Sal_Relational_Operator = 0; }</pre>	<pre>(AssignmentOfConditions &gt; 0) Sal_Relational_Operator = (a &gt; 0);</pre>

**Optimization = ON**

≤ TargetLink 3.5	TargetLink 4.0
<pre>(NoAssignmentOfBooleanIfThenElse=OFF) (default) Sal_Relational_Operator = (a &gt; 0);  — or — (dependent on the data type of a) Sal_Logical_Operator = (a != 0);</pre>	<pre>Sal_Relational_Operator = (a &gt; 0);</pre>

This change also affects logical or relational operations from other sources. By default, TargetLink now only transforms the non-optimized control flow version into an assignment if the assignee is a Boolean or bitfield variable.

```

Sa1_Logical_Operator1 = b && c;
if (Sa1_Logical_Operator1 != 0) {
    Sa1_BoolSwitch = 0;
} else {
    Sa1_BoolSwitch = 1;
}

Sa1_Logical_Operator2 = d <= 0;
if (Sa1_Logical_Operator2 != 0) {
    Sa1_Int8Switch = 0;
} else {
    Sa1_Int8Switch = 1;
}

Sa1_Logical_Operator3 = e != 0;
if (Sa1_Logical_Operator3 != 0) {
    Sa1_FloatSwitch = 0.F;
} else {
    Sa1_FloatSwitch = 1.F;
}
    
```

≤ TargetLink 3.5	TargetLink 4.0
(NoAssignmentOfBooleanIfThenElse=OFF (default)) Sa1_BoolSwitch = (!b    !c);  Sa1_Int8Switch = (d > 0);  Sa1_FloatSwitch = (e == 0);	(AssignmentOfConditions=2 (BooleanOutputsOnly; default)) Sa1_BoolSwitch = (!b    !c);  if (d <= 0) { Sa1_Int8Switch = 0; } else { Sa1_Int8Switch = 1; }  if (e != 0) { Sa1_FloatSwitch = 0.F; } else { Sa1_FloatSwitch = 1.F; }  (AssignmentOfConditions=3 (All Outputs)) Sa1_BoolSwitch = (!b    !c);  Sa1_Int8Switch = (Int8) (d > 0);  Sa1_FloatSwitch = (Float32) (e == 0);

This change in the initial code pattern can lead to different optimization results in the generated code, e.g., assignments involving a logical or relational operation take place later than with TargetLink ≤3.5.

**RDI macro definitions**

TargetLink now places brackets around the initial values that receive a cast. This also means improved MISRA-C compliance.

≤ TargetLink 3.5	TargetLink 4.0
<code>#define PIM_RDI (sint32 (*)(2)) Rte_Pim_PIM_Matrix3x2()</code>	<code>#define PIM_RDI ((sint32 (*)(2)) Rte_Pim_PIM_Matrix3x2())</code>

**Improved code efficiency changes**

With TargetLink 4.0, code efficiency was improved, which can also lead to changes in the generated code compared to older TargetLink versions. For details, refer to *Improved Code Efficiency* on page 166.

## Code Generator Options

### Migration Aspects Regarding Code Generator Options

**Renamed Code Generator options**

The following Code Generator options were renamed:

Old Name	New Name
NoAssignmentOfBooleanExpressions	AssignmentOfConditions
PolySpaceSupport	InsertComputeThroughOverflowComments
DisableArbitraryOptimizations	Allow64BitMultiplicationsForArbitraryScaled16BitOperands

**Recommended compatibility settings for new Code Generator options**

For the best downward compatibility to TargetLink 3.5, it is recommended to use the *Compatibility Setting* values for the new Code Generator options listed in the following table:

Code Generator Option	Compatibility Setting	Default Value
AssignmentOfConditions	3 - AllOutputs <sup>1)</sup> 0 - None <sup>2)</sup>	2 - AllBooleanOutputs
Allow64BitMultiplicationsForArbitraryScaled16BitOperands <sup>3)</sup>	on <sup>4)</sup> off <sup>5)</sup>	on

<sup>1)</sup> When Optimization option is set to on.

<sup>2)</sup> When Optimization option is set to off.

<sup>3)</sup> Substitutes DisableArbitraryOptimizations

<sup>4)</sup> When DisableArbitraryOptimizations was set to off in TargetLink 3.5.

<sup>5)</sup> When DisableArbitraryOptimizations was set to on in TargetLink 3.5.



For TargetLink versions < 3.5, refer to the versions' *New Features and Migration* document. For details, refer to *Previous release documents* on page 34.

**Removed Code Generator option**

The following Code Generator option was removed from TargetLink:

Removed Option	Replacement Option	Compatibility Setting
EfficientVectorHandling	LoopUnrollThreshold ( <a href="#">TargetLink Block and Object Reference</a> ) <sup>1)</sup> .	INF, if EfficientVectorHandling was set to off

<sup>1)</sup> For details, refer to *Removal of EfficientVectorHandling* on page 197 and *Basics on Processing Vectors and Matrices* ([TargetLink Preparation and Simulation Guide](#)).

**New Code Generator options**

For more information on new Code Generator options, refer to *New Code Generator Options* on page 177.

**Related topics**

References

- *Code Generator Options* ([TargetLink Block and Object Reference](#))

## Changes With Access Functions

**Where to go from here**

Information in this section

<i>New Default Macro Bodies for Macro Access Functions</i>	201
<i>Changes of ADDRESS_BY_PARAMETER Access Functions</i>	202
<i>New Access-Function-Specific Name Macro</i>	203

## New Default Macro Bodies for Macro Access Functions

TargetLink provides default macro bodies for macro access functions. These defaults are used in the following circumstances:

- The DD AccessFunction object references a DD FunctionClass object whose Macro property is set to on.
- The DD AccessFunction object's MacroBody property is empty.

To increase MISRA-C compliance, the following placeholders for macro arguments are enclosed by brackets:

- `_var`
- `_value`

The example macro bodies of the predefined access function templates contained in the `dsdd_master_advanced` and `dsdd_master_autosar3.dd [System]/dsdd_master_autosar4.dd [System]` Data Dictionary Manager system templates were adapted accordingly.

#### Related Documentation

- *Basics on Access Functions* ([📖 TargetLink Customization and Optimization Guide](#))
- *Specifying Access Functions as Preprocessor Macros* ([📖 TargetLink Customization and Optimization Guide](#))
- *Overview of Predefined Access Function Templates* ([📖 TargetLink Customization and Optimization Guide](#))

## Changes of ADDRESS\_BY\_PARAMETER Access Functions

The type of the `_var` macro argument has changed from `scalar type` to `pointer to scalar type` for *vector variables*. This is the case in the following circumstances:

- The DD `AccessFunction` object has its `Kind` property set to `ADDRESS_BY_PARAMETER`.
- The DD `AccessFunction` object references a DD `FunctionClass` object whose `Macro` property is set to `on`.

TargetLink 3.5	TargetLink 4.0
<code>(GetAddressU16(<b>vector</b>[0])) [1] = 5;</code>	<code>(GetAddressU16(<b>vector</b>)) [1] = 5;</code>



With TargetLink 4.0 you cannot use the same macro body to access scalars and vectors but have to use a separate macro body per variable kind.

When upgrading an existing Data Dictionary, TargetLink sets *all* empty `Kind` properties to `APPLY_TO_SCALAR`. This affects *both* macro AFs *and* function AFs.



If you use TargetLink's predefined access function templates, you can copy the new versions from a DD workspace based on the dsdd\_master\_advanced/dsdd\_master\_autosar3.dd [System]/dsdd\_master\_autosar4.dd [System] system templates.

## New Access-Function-Specific Name Macro

TargetLink now has the following new name macros:

Name Macro	Description
\$(Dim1Width), \$(Dim2Width)	<p>These name macros are allowed in access function templates to specify function names and macro bodies:</p> <ul style="list-style-type: none"> <li>■ \$(Dim1Width) - Replaced by the number of elements in the 2-D variable's first dimension (row).</li> <li>■ \$(Dim2Width) - Replaced by the number of elements in the 2-D variable's second dimension (column).</li> </ul> <p>No effect for scalar variables. For vector variables, only \$(Dim1Width) is applicable.</p>
\$(VarAccess)	<p>This macro is allowed in access function templates to specify macro bodies and gives correct access to the variable concerned, as and when required: For non-struct component variables, it is replaced in the same way as \$v, for struct component variables, it is replaced in the same way as \$(Sv) \$(SPath) . \$v.</p>

### Related documentation

- *Basics on Using Name Macros* ([📖 TargetLink Customization and Optimization Guide](#))
- *Changing Access Function Implementations Using Name Macros* ([📖 TargetLink Customization and Optimization Guide](#))
- *Overview of Predefined Access Function Templates* ([📖 TargetLink Customization and Optimization Guide](#))

## AUTOSAR-Related Migration Aspects

### Where to go from here

Information in this section

<i>2-D Matrix Data Elements and Operation Arguments</i>	204
<i>AUTOSAR Export</i>	205
<i>Separate Installation of Container Manager</i>	205
<i>Application Data Types</i>	206
<i>Constraints of Array and Matrix Types</i>	207
<i>Replaced IsQueued Property</i>	207

## 2-D Matrix Data Elements and Operation Arguments

TargetLink supports passing array types via a pointer to the array's first scalar element, as described by the AUTOSAR standard. This holds for parameters and return values of the RTE API.

**Code examples:** To improve code readability, TargetLink locally introduces the following auxiliary pointers:

### 2-D array passed to RTE API function

```
Rte_Write_SP_MyDataElem(&A[0][0]);
```

### 2-D array passed to non-RTE API function

```
CalcSRV(A);
```

### 2-D array with base type sint16 and 8 columns returned by RTE API function

```
p_MyDataElem =
    (const sint16 (*) [8]) Rte_IRead_Run_RP_MyDataElem();
...
Sa4_Selector = p_MyDataElem[IndexX][IndexY];
```

### 2-D array with non-RTE API access function of kind ADDRESS

```
Sa4_Selector = (GetAddressOfA()) [IndexX][IndexY];
```

## AUTOSAR Export

### Version-specific export

With the introduction of `IncludedDataSet`s in the AUTOSAR 4.x standard, the code representation in the Data Dictionary `Subsystems` area depends on the AUTOSAR version the code was generated for.

Accordingly, AUTOSAR export from the Data Dictionary obeys the following compatibility matrix:

Code Generated for Version	AR 2.x Export	AR 3.x Export	AR 4.x Export
AUTOSAR 2.x	Possible	Possible	Not possible
AUTOSAR 3.x	Possible	Possible	Not possible
AUTOSAR 4.x	Not possible	Not possible	Possible

### Export of 2-D arrays

For AUTOSAR 4.x, data types for 2-D arrays are now exported as nested implementation data types.

## Separate Installation of Container Manager

dSPACE's Container Manager is now separately installed and will remain on your system if you remove TargetLink.

**Providing a custom workflow definition (CTW) file** You can provide a custom workflow definition file to change workflow rules.



It is recommended that only experienced users change the workflow rules. There is a special syntax for defining workflow rules.

1. Provide a copy of the CTW template via TargetLink's `tlCustomizationFiles` API function and save it to `<MyDir>`.
2. Change the CTW file's workflow rules as required.
3. In the Data Dictionary Manager, provide the path to `<MyDir>\<CTW file name>` as the value of the `WorkflowDefinitionFile` property contained in the `/Pool/Autosar/Config/ContainerExchange/` option set or as the value of the `tl_export_container` API function's `WorkflowDefinitionFile` property.

### Related documentation

- *Preparing Container Export in TargetLink* ( Container Management Document)
- *Advanced: Configuring Container Handling* ( Container Management Document)

### Related topics

#### References

- *tl\_export\_container* ( TargetLink API Reference)

## Application Data Types

With TargetLink 4.0 the following changes were made according to the AUTOSAR standard:

The `InvalidValue`, `ScalingRef`, and `UnitRef` properties can only be set at `ApplicationDataType` objects (ADTs) whose `Kind` property is set to `Primitive`.

DD Object	DD Property		
	<code>InvalidValue</code>	<code>ScalingRef</code>	<code>UnitRef</code>
ApplicationDataType objects whose Kind property is set to <code>Array</code> or <code>Record</code>	Forbidden	Forbidden	Forbidden
ApplicationDataTypeComponent	Removed	Removed	Removed

**Consistency check** During the upgrade of a legacy Data Dictionary to the latest data model revision, TargetLink performs a consistency check:

Array ADTs	Record ADTs
<ol style="list-style-type: none"> <li>1. TargetLink checks whether the values of the <code>InvalidValue</code>, <code>ScalingRef</code>, and <code>UnitRef</code> properties found at an array ADT match the values of these properties at the primitive ADT that is referenced at the ADT.</li> <li>2. If the values match, the <code>InvalidValue</code>, <code>ScalingRef</code>, and <code>UnitRef</code> properties are unset at the array ADT.</li> <li>3. If the values do not match, these properties are left untouched.</li> </ol> <p>An error occurs during validation.</p>	<ol style="list-style-type: none"> <li>1. TargetLink checks whether the values of the <code>InvalidValue</code>, <code>ScalingRef</code>, and <code>UnitRef</code> properties found at each of the record ADT <code>ApplicationDataTypeComponent</code> objects match the values of the primitive ADT referenced at each component.</li> <li>2. If the values match, the <code>InvalidValue</code>, <code>ScalingRef</code>, and <code>UnitRef</code> properties are removed from each component.</li> <li>3. If the values do not match, these properties remain at each component as custom properties.</li> </ol> <p>No error occurs during validation.</p>



If TargetLink detects inconsistencies, resolve the conflicts manually to provide a specification basis according to the AUTOSAR standard.

## Constraints of Array and Matrix Types

For AUTOSAR array/matrix types (i.e., types whose `AutosarArrayWidth` is  $> 0$ ) that are used with AUTOSAR interfaces, you no longer have to create a `Constraints` object. During code generation, TargetLink draws on the constraints defined at the primitive data type(s).

You still have to create a `Constraints` object for array/matrix types, if:

- The type is used at a non-AUTOSAR interface.
- The type is used in Standard code generation mode.

## Replaced IsQueued Property

For `DataElement` objects, the following change was made:

Removed Property	Replacement	Compatibility Setting
<code>IsQueued</code>	<code>ImplementationPolicy</code>	standard, if <code>IsQueued</code> was set to <code>off</code> queued if <code>IsQueued</code> was set to <code>on</code>



Data Dictionary files created with TargetLink versions prior to TargetLink 4.0 are automatically migrated.

## Other

### Where to go from here

Information in this section

<i>Various Migration Aspects</i>	208
<i>Stricter Settings for Bus Diagnostics</i>	209
<i>Plot Channel Specification</i>	211
<i>Signal Properties Inheritance</i>	212

## Various Migration Aspects

### OpenFcn callbacks

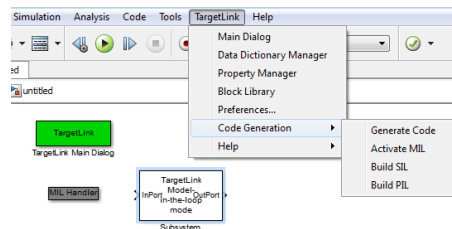
Simulink does not allow using OpenFcn callbacks that directly or indirectly change the Commented block property. Because TargetLink 4.0 uses the Commented block property when switching the TargetLink subsystem to SIL or PIL simulation mode, custom buttons created, for example, to start code generation no longer work (*Improvements to TargetLink's Simulation Frame* on page 161).

The following API functions change the Commented block property and cannot be used directly or indirectly in OpenFcn callbacks:

- `tl_generate_code`
- `tl_build_host`
- `tl_build_target`
- `tl_set_sim_mode`

**Solution** Several workarounds are possible:

- Use the following *TargetLink Utility Blocks* ([TargetLink Block and Object Reference](#)):
  - MIL mode (refer to *MIL Mode Block* ([TargetLink Block and Object Reference](#))) block
  - SIL mode (refer to *SIL Mode Block* ([TargetLink Block and Object Reference](#))) block
  - Tool Selector (refer to *Tool Selector Block* ([TargetLink Block and Object Reference](#))) block
- Use the Model Window's TargetLink menu:



- Use the controls of the TargetLink Main Dialog (refer to *Code Generation Page (Main Dialog Block)* ([TargetLink Block and Object Reference](#))) block.



- Use annotations with ClickFcn callbacks:
  1. Double-click an empty area within the Simulink Model Window to produce an annotation.
  2. Enter text as required.
  3. Select the annotation and select **Properties...** from its context menu. The **Annotation properties** dialog opens.
  4. In the **ClickFcn** group box, specify the command to be run by the annotation.
  5. In the **Appearance** group box, specify a button-like appearance for the annotation.
  6. Click **OK** to close the dialog.

You can now run your command by single-clicking the annotation.
- Write your own GUI with controls that start the code generation process.
- Create your own Simulink menu to start your scripts as described in the Simulink Documentation

## Stricter Settings for Bus Diagnostics

### Automatic configuration adaptation

TargetLink might automatically change the values of some Simulink configuration parameters in the following circumstances:

- During model preparation
- During model upgrade
- When a TargetLink Main Dialog block is added to a model
- When a MIL Handler block is added to a model

This lets you use the full range of TargetLink MIL simulation features such as logging and overflow detection.

Simulink Configuration Parameter	Value Set By TargetLink
Element name mismatch	error <sup>1)</sup>
Mux blocks used to create bus signals	error <sup>1)</sup>
Bus signal treated as vector	error <sup>1)</sup>
Non-bus signals treated as bus signals	error <sup>1)</sup>
Signal storage reuse	off

Simulink Configuration Parameter	Value Set By TargetLink
Block reduction	off
Signal logging format	Dataset

<sup>1)</sup> If you reset this preference to a value other than 'error', TargetLink displays message E02462 at simulation start or during code generation.

### Required model parameters

Several Simulink Model Configuration parameters located in **Diagnostics - Connectivity - Buses** must be set according to the following table:

Parameter	Setting
Element name mismatch	error
Mux blocks used to create bus signals	error
Bus signal treated as vector	error
Non-bus signals treated as bus signals	error

TargetLink checks for these parameter settings during model initialization. If these settings do not exist, TargetLink displays error message E02462 and code generation is not possible.

The following TargetLink MIL simulation capabilities are then locked as well and message E02464 will be displayed:

- Signal logging
- Min/max logging
- Signal plotting
- Overflow detectoin

**API** To set these properties via the API using Simulink's `get_param/set_param` API functions, make the following settings:

Property	Value
StrictBusMsg	'ErrorOnBusTreatedAsVector'
NonBusSignalsTreatedAsBus	'error'
BusObjectLabelMismatch	'error'



Simulink's Bus to Vector block conflicts with the bus signal treated as vector and non-bus signals treated as bus signal parameter settings. Do not use this block.

## Plot Channel Specification

### New method of plot channel specification

For matrix signals, TargetLink provides two ways to specify plot channels:

- Linear indexing of matrix signal elements (in column major order)
- One-based index pair for each matrix signal element ([row column]).



We recommend to specify the plot channels of matrix signals using the index pair approach.

**Example of the index pair approach** The plot channels [1, 4; 2, 3; 5, 6] specify to plot the three matrix elements (1, 4), (2, 3), and (5, 6) during simulation.

### Migration consideration



For matrix signals, the specification of a plot channel containing *exactly one* pair of one-based indices is ambiguous because it can stand for one of the following:

- Two matrix signal elements (linear index column-wise)
- One matrix signal element (index pair row/column).

TargetLink resolves this ambiguity by interpreting such a plot channel specification as being an index pair.

For models created with a TargetLink version  $\leq 3.4$ , a problem can occur if the model contains row matrices ( $[1 \times n]$ ) or column matrices ( $[m \times 1]$ ):

Problem	Description
One signal plotted instead of two	TargetLink blocks connected to row/column matrix signals plot one instead of two signals.
Invalid plot channels	TargetLink detects invalid plot channels and displays warning W02441 ( <a href="#">TargetLink Message Reference</a> ).

**Example of invalid plot channels** A  $[6 \times 1]$  matrix signal is connected to a TargetLink block whose plot channels are defined as [3 5].

### Solution

Often, the problems result from signals originating from InPort blocks or Constant blocks:

Source	Solution
InPort block	Explicitly specify the signal's dimension to make it a vector.
Constant block	Select the Interpret as 1-D checkbox.

## Signal Properties Inheritance

TargetLink's code generator no longer allows to mux or concatenate several signals of different data types if these types have different base types.

**Example** Block A and block B have output variables of different data types. Their output signals are muxed and fed into block C, whose signal inheritance is enabled:

TargetLink ≤ 3.5	TargetLink 4.0
Two output variables for block C, each having one of the inherited data types	E15738 ( <a href="#">TargetLink Message Reference</a> )

**Solution** Use a Bus Creator block instead of a Mux block.

## API Commands

### Changes in TargetLink and TargetLink Data Dictionary API Functions

#### Renamed API commands

The following TargetLink API commands were replaced:

TargetLink ≤3.5	TargetLink 4.0 <sup>1)</sup>
<code>tl_extract_subsystem</code>	<code>tlExtractSubsystem</code> ( <a href="#">TargetLink API Reference</a> )
<code>tl_sim_interface</code>	<code>tlSimInterface</code> ( <a href="#">TargetLink API Reference</a> )
<code>tl_switch_blockset</code>	<code>tlOperationMode</code> ( <a href="#">TargetLink API Reference</a> )
<code>tl_upgrade</code>	<code>tlUpgrade</code> ( <a href="#">TargetLink API Reference</a> )

<sup>1)</sup> The commands are case-sensitive.

## Obsolete

### Where to go from here

Information in this section

<i>Discontinued TargetLink Features</i>	213
<i>Discontinued Data Dictionary Features</i>	213
<i>Obsolete Limitations</i>	214
<i>Obsolete API Functions</i>	214

## Discontinued TargetLink Features

### Discontinued TOMs

As of this TargetLink version, the following TOMs are no longer available:

- HCS12:  
HCS12/Cosmic 4.7  
HCS12/CodeWarrior 3.1
- MPC5xx:  
MPC5xx/Green Hills 5.1  
MPC5xx/WindRiver Diab 5.7
- MPC55xx:  
MPC55xx/WindRiver Diab 5.7
- M32R:  
M32R/GAIO 9

## Discontinued Data Dictionary Features

### Use Case filter in the Data Dictionary Manager

The Use Case filter function is no longer available in the Data Dictionary Manager. You can now use and specify your own XML-based filter rule sets. For further information, refer to *Improvements to the Data Dictionary* on page 168.

## Obsolete Limitations

With TargetLink 4.0, the following limitations of previous TargetLink versions were removed:

### General limitations

#### Multidimensional signals

TargetLink does not support multidimensional signals (matrix signals) at block outputs. Matrix parameters are supported only for Look-Up Table (2-D), Interpolation Using Prelookup, Direct Look-Up Table (n-D), Discrete State-Space, and Custom Code blocks. Parameters with more than two dimensions are not possible.

In Stateflow, the interface cannot contain multidimensional signals, but local variables can be matrices. For details on Stateflow limitations, refer to *Stateflow Limitations* ([📖 TargetLink Orientation and Overview Guide](#)).



This TargetLink version now supports 2-D matrix signals. However, 3-D signals (or higher) are not supported.

### Block-specific limitations

#### Gain block

These blocks are only supported by TargetLink when the Multiplication parameter in the Simulink block dialog is set to Element-wise(K.\*u) (default).

### TargetLink AUTOSAR

#### Module Limitations

#### Model-linked code view

TargetLink does not support the model-linked code view for blocks with activated AUTOSAR communication.

#### Updating frame model / Name modifications

When updating a generated frame model from AUTOSAR data, TargetLink treats renamed AUTOSAR elements as new. TargetLink prompts you to remove the blocks carrying the old names from the model and adds the blocks with the new names.

## Obsolete API Functions

Function	Status	Replacement
build_customcode_sfcn	Error <sup>1)</sup>	<code>tl_build_customcode_sfcn</code>
get_mdltldata	Error <sup>1)</sup>	-
get_sfobjects	Error <sup>1)</sup>	<code>tl_get_sfobjects</code>
get_tlblocks	Error <sup>1)</sup>	<code>tl_get_blocks</code>
tl_adapt_dd_references	Warning <sup>2)</sup>	<code>tlMoveDDObject</code>

Function	Status	Replacement
tl_extract_subsystem	Warning <sup>2)</sup>	<b>tlExtractSubsystem</b>
tl_find_system	Error <sup>1)</sup>	find_system
tl_get_blockset_mode	Warning <sup>2)</sup>	<b>tlOperationMode</b>
tl_get_project	Error <sup>1)</sup>	dsdd_manage_project('GetProjectFile', ...)
tl_set_project	Error <sup>1)</sup>	dsdd_manage_project('SetProjectFile', ...)
tl_sim_interface	Warning <sup>2)</sup>	<b>tlSimInterface</b>
tl_switch_blockset	Warning <sup>2)</sup>	<b>tlOperationMode</b>

<sup>1)</sup> The function was removed from TargetLink.

<sup>2)</sup> The function is obsolete and will be removed in a future version of TargetLink.

**Compatibility consideration** Adapt your user scripts and toolchain accordingly.

## Messages

### Message Changes

#### Changed message type

The following messages changed their type to `Advice` in TargetLink 4.0 because they only inform you about possible code efficiency improvements, but can be safely ignored:

Old Message Number	New Message Number
W15658	A15658
N17350	A17350
W17352	A17352
W17358	A17358
W17356	A17356

## Stateflow-Related Changes

---

**Where to go from here**

Information in this section

<i>Exported Graphical Functions</i>	216
<i>Stateflow Matrices</i>	217

---

## Exported Graphical Functions

---

**Simulation in SIL/PIL not possible**

Due to *improvements in TargetLink's simulation frame* (refer to *Improvements to TargetLink's Simulation Frame* on page 161), Exported Graphical Functions used outside of but defined within the TargetLink subsystem cannot be simulated in SIL or PIL simulation mode.

This is the case because the TargetLink subsystem is commented in SIL or PIL simulation mode.

---

**Code generation**

TargetLink can generate production code for Exported Graphical Functions.

Stateflow Coder cannot build Exported Graphical Functions defined in commented model parts. Accordingly, it cannot build Exported Graphical Functions defined in a TargetLink subsystem set to SIL or PIL simulation mode.

---

**Modeling recommendation**

If you solely want to use Exported Graphical Functions *outside* of TargetLink subsystems, do *not* define them *within* TargetLink subsystems. This avoids dead code.



## Stateflow Matrices

### Loop generation

TargetLink now uses the same mechanisms and optimizations for Stateflow and Simulink matrices. Accordingly, both dimensions can now be rolled into loops in the generated production code.

TargetLink 4.0	TargetLink 3.x
Outer dimension's iteration generated as <code>for</code> loop Rolling depends on <code>LoopUnrollThreshold</code>	Outer dimension's iteration generated as <code>do-while</code> loop Not subject to <code>LoopUnrollThreshold</code>

### Possible code changes

- Merged loops
- Other auxiliary variables for vectors/matrices
- Former `do while` loops can now be `for` loops.

### Related documentation

- *Code Pattern for Vectors and Matrices* ([📖 TargetLink Preparation and Simulation Guide](#))
- *Merging consecutive loops* ([📖 TargetLink Customization and Optimization Guide](#))

# Changes in Future TargetLink Versions

## To be Discontinued

---

**A2L Import**

It is planned to discontinue the A2L import in a future TargetLink version.

---

**Generation of RTF documents**

It is planned to discontinue the option to generate documentation in rich text format (RTF) in a future TargetLink version.

# VEOS

---

## Where to go from here

Information in this section

<i>New Features of VEOS 3.3</i>	219
<i>Migrating to VEOS 3.3</i>	220

## New Features of VEOS 3.3

---

### Support of LIN bus simulation

VEOS supports the following features in a LIN bus simulation:

- Simulation of LIN slave nodes  
The same ECU parameter definition is used for generating LIN master nodes and LIN slave nodes.
- Simulation of multiple LIN master and slave nodes on one V-ECU
- Simulation of the communication between LIN slave nodes
- Simulation of LIN message collisions on the bus
- Simulation compliant with LIN bus transfer times
- Simulation of broadcast frames

---

### Stimulating the variables of V-ECUs in offline simulation

The variables of V-ECUs can now be stimulated in offline simulation on VEOS by using ControlDesk's Signal Editor. See also *New Signal Editor Features (ControlDesk 5.3)* on page 91.

**Supported compilers**

VEOS 3.3 now supports Microsoft Visual C/C++ Compiler (MSVC) 11 (provided by Microsoft Windows® SDK 8.1).



The support of MSVC 9.0 is discontinued in VEOS 3.3.

## Migrating to VEOS 3.3

**Compatibility overview**

The table shows the compatibility between VEOS and OSA files and V-ECU implementations:

	OSA Files Created with Products of dSPACE Release ...			V-ECU Implementations Created with Products of dSPACE Release ...		
	2013-B and Earlier	2014-A	2014-B	2013-B and Earlier <sup>1)</sup>	2014-A <sup>2)</sup>	2014-B <sup>3)</sup>
VEOS 3.2	– <sup>4)</sup>	✓ <sup>5)</sup>	–	✓	✓	–
VEOS 3.3	– <sup>4)</sup>	✓	✓	✓	✓	✓

<sup>1)</sup> V-ECU implementation version 1.0

<sup>2)</sup> V-ECU implementation version 2.0

<sup>3)</sup> V-ECU implementation version 2.1

<sup>4)</sup> For migration information, refer to the *New Features and Migration* document for dSPACE Release 2014-A. You can download the document from [www.dspace.com/go/migration](http://www.dspace.com/go/migration).

<sup>5)</sup> An OSA file created with a product of dSPACE Release 2014-A and modified with VEOS 3.3 cannot be loaded with VEOS 3.2.

---

# Compatibility Information

---

**Where to go from here**

Information in this section

<i>Supported MATLAB Releases</i>	222
<i>Operating System</i>	224
<i>Run-Time Compatibility of dSPACE Software</i>	226
<i>Limitations for 64-Bit Windows Operating Systems in Combination with 32-Bit dSPACE Software</i>	227
<i>Limitations for Products on 64-Bit dSPACE DVD</i>	227
<i>Limitations for Windows 7</i>	230

## Supported MATLAB Releases

Supported MATLAB releases

MATLAB Release...	...Is Supported by dSPACE Release 2014-B						
	RCP and HIL Software						
	RCP and HIL Software from 32-bit DVD with 32-bit MATLAB versions	RCP and HIL Software from 64-bit DVD with 64-bit MATLAB versions	AutomationDesk 4.1 <sup>1)2)</sup>	TargetLink 4.0	Model Compare 2.5	VEOS 3.3	dSPACE Python Extensions 1.7 <sup>3)2)</sup>
R2014b	✓	✓	✓	✓	✓	✓	✓
R2014a	✓	✓	✓	✓	✓	✓	✓
R2013b	✓ <sup>4)</sup>	✓ <sup>4)</sup>	✓	✓	✓	✓	✓
R2013a	✓ <sup>5)</sup>	✓ <sup>5)</sup>	✓	✓	✓	✓	✓

<sup>1)</sup> AutomationDesk's MATLAB Access library requires MATLAB.

<sup>2)</sup> The 32-bit version also supports 64-bit MATLAB versions if the 32-bit MATLAB version from the same MATLAB release is installed on your host PC in parallel. For details, see the notes below this table.

<sup>3)</sup> matlablib2 of dSPACE Python Extensions requires MATLAB.

<sup>4)</sup> Not supported by the RTI FPGA Programming Blockset-FPGA Interface.

<sup>5)</sup> R2013a: Due to performance problems with dSPACE Automotive Simulation Models (ASMs), it is recommended to install the following bug fix before you use ASMs with R2012b or R2013a: <http://www.mathworks.com/support/bugreports/916069>.

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/sw3rdparty>.

**Support of MATLAB R2014a** With MATLAB R2014a, code generation by Simulink Coder has changed, so the generation of variable descriptions (TRC files) by RTI, ConfigurationDesk and VEOS has to be modified accordingly. These changes cannot fully be taken into account in dSPACE Release 2014-A. As a consequence, dSPACE Release 2014-A does *not officially* support MATLAB R2014a. There is a special MATLAB command, however, that lets you enable the previous code and TRC file generation behavior. This way, dSPACE Release 2014-A unofficially supports MATLAB R2014a. For details, refer to <http://www.dspace.com/go/MATLABR2014a>.

## Operating System

**Operating system on host PC**

The following operating systems are supported by the dSPACE products on Release 2014-B:

32-Bit dSPACE Software	64-Bit dSPACE Software
<ul style="list-style-type: none"> <li>■ Windows 7 Professional, Ultimate, and Enterprise with Service Pack 1 (32-bit or 64-bit version) Only the listed editions are supported. The Windows 7 Home and Starter editions are not supported.</li> </ul>	<ul style="list-style-type: none"> <li>■ Windows 7 Professional, Ultimate, and Enterprise with Service Pack 1 (64-bit version) Only the listed editions are supported. The Windows 7 Home and Starter editions are not supported.</li> </ul>
<b>Notes and Limitations</b>	
<ul style="list-style-type: none"> <li>■ Limitations apply when you use Windows 7. Refer to <i>Limitations for Windows 7</i> on page 230.</li> <li>■ Support of 64-bit operating systems: 32-bit dSPACE software supports only the 64-bit version of Windows 7. Other 64-bit operating systems (Windows XP and Windows Vista) are not supported. 32-bit dSPACE software runs under 64-bit Windows operating systems in a WoW64 (Windows-on-Windows 64-bit) subsystem. Limitations apply when you use a 64-bit Windows operating system with 32-bit dSPACE software. Refer to <i>Limitations for 64-Bit Windows Operating Systems in Combination with 32-Bit dSPACE Software</i> on page 227.</li> <li>■ ControlDesk Next Generation can also be installed on the MicroAutoBox Embedded PC, running under Microsoft Windows 7 Ultimate (32-bit version).</li> </ul>	<ul style="list-style-type: none"> <li>■ Limitations apply when you use Windows 7. Refer to <i>Limitations for Windows 7</i> on page 230.</li> <li>■ Only TargetLink, Model Compare, and various products of the RCP and HIL software package are available as 64-bit versions. Limitations apply when you use these 64-bit versions. For details, refer to <i>Limitations for Products on 64-Bit dSPACE DVD</i> on page 227.</li> </ul>

**32-bit dSPACE software running on 64-bit Windows operating systems** dSPACE software runs as a 32-bit application under 64-bit Windows operating systems in a WoW64 (Windows-on-Windows 64-bit) subsystem. WoW64 is the x86 emulator of Windows that allows 32-bit Windows-based applications to run seamlessly on 64-bit versions of Windows. This lets you use up to 4 GB of virtual memory for each 32-bit process if the application is prepared for using the large memory area. Otherwise, the virtual address space of a process is limited to 2 GB.



### Allowing communication via 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, and 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="%dSPACE_root%\MotionDesk\Bin\MotionDesk.exe"
dir=in action=allow profile=any description="Allow dSPACE MotionDesk to receive motion data via network."
```

If you are running third-party firewall software on your host PC, ensure that the TCP/IP communication of dSPACE software is not blocked.

### Operating system on dSPACE License Server

If you purchased floating network licenses, you have to install and configure one of the networked PCs as the dSPACE License Server. The operating system of the dSPACE License Server must be one of the following:

- Windows XP Professional (32-bit version) with Service Pack 3
- Windows Vista Business, Ultimate, or Enterprise (32-bit or 64-bit version) with the latest Service Pack
- Windows 7 Professional, Ultimate, or Enterprise (32-bit or 64-bit version) with the latest Service Pack
- Windows Server 2003 (32-bit or 64-bit version)
- Windows Server 2008 R2
- Windows Server 2012



The dSPACE License Server does not support non-Windows operating systems.

## 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 2014-B

dSPACE recommends to use only software products from the same dSPACE Release to provide maximum run-time compatibility.

Note that:

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

If dSPACE products interact directly (for example, through automation interfaces) or indirectly (for example, through common file types like A2L), limitations may 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 information on whether a patch is necessary and on the patch itself, refer to <http://www.dspace.com/go/CompPatch>.

- RCP and HIL software products (on Release 2014-B) cannot be used in combination with RCP and HIL software products from earlier dSPACE releases.

**Major limitations for TargetLink and Model Compare** The 64-bit version of TargetLink cannot be used in combination with the 32-bit version of Model Compare and vice versa because you can work only with a bit-compatible MATLAB version (32-bit or 64-bit).

**Major limitation for working with a SCALEXIO system** The products for working with a SCALEXIO system must be compatible. This is only guaranteed for products delivered with the same dSPACE Release. Contact dSPACE for further information if you have a query.

---

### Combining dSPACE products from earlier releases

For detailed information and notes on the combined use of different products from and with earlier releases, refer to [http://www.dspace.com/go/ds\\_sw\\_combi](http://www.dspace.com/go/ds_sw_combi).

## Limitations for 64-Bit Windows Operating Systems in Combination with 32-Bit dSPACE Software


<b>Objective</b>	Some additional limitations apply when you use 64-bit versions of Windows 7 in combination with 32-bit dSPACE software.
<b>Limitations of device drivers</b>	Third-party bus interfaces (CAN, LIN, or FlexRay) are supported only if they have 64-bit drivers from the manufacturers.
<b>TargetLink: Limitations of target compilers</b>	For information on support for a specific target compiler, contact the respective compiler manufacturer.
<b>MATLAB</b>	If you install a 32-bit version of MATLAB under Windows 7 (64-bit versions), the MATLAB installation program generates a message that a 64-bit version of MATLAB is available. To install the 32-bit version of MATLAB, click OK.

## Limitations for Products on 64-Bit dSPACE DVD

<b>Objective</b>	<p>In general, the 64-bit dSPACE DVD set contains the same products as the 32-bit dSPACE DVD set. However, the 64-bit DVD set contains:</p> <ul style="list-style-type: none"> <li>■ All MATLAB-related dSPACE products which have been ported to support 64-bit MATLAB versions</li> <li>■ All 32-bit dSPACE products which also supports 64-bit MATLAB versions.</li> <li>■ All 32-bit dSPACE products that do not relate to MATLAB (e.g., ControlDesk Next Generation).</li> </ul> <p>When using the 64-bit DVD set, you should also note the limitations described below.</p>
------------------	---

### dSPACE software products ported to 64-bit

The following table lists all dSPACE products which support MATLAB and their availability as 64-bit versions on the 64-bit dSPACE DVD set.

	Keep in mind that dSPACE products without any MATLAB support, e.g., ControlDesk Next Generation, are contained only as 32-bit version.
---	--

dSPACE Products with MATLAB Support		Contained on 64-Bit DVD set as ...	
		64-Bit Version	32-Bit Version
AutomationDesk		–	✓ <sup>1)</sup>
TargetLink		✓	–
Model Compare		✓	–
VEOS		–	✓ <sup>2)</sup>
Platform API Package	dSPACE Python Extensions	–	✓ <sup>1)</sup>
	HIL API .NET	–	✓
	XIL API .NET 2.0 MAPort	–	✓
RCP and HIL software package	RTI and RTI-MP	✓	–
	RTI Gigalink Blockset	✓	–
	RTI CAN Blockset	✓	–
	RTI CAN MultiMessage Blockset	✓	–
	RTI LIN MultiMessage Blockset	✓	–
	RTI FlexRay Configuration Blockset	✓	–
	RTI FPGA Programming Blockset	✓	–
	RTI Electric Motor Control Blockset	✓	–
	RTI Ethernet Blockset	✓	–
	RTI Ethernet UDP Blockset	✓	–
	RTI XCP on Ethernet Blockset	✓	–
	RTI Watchdog Blockset	✓	–
	RTI RapidPro Control Unit Blockset	✓	–
	RTI Bypass Blockset	✓	–
	RTI USB Flight Recorder Blockset	✓	–
	ConfigurationDesk	–	✓ <sup>2)</sup>
	FlexRay Configuration Blockset	✓	–
	FlexRay Configuration Tool	–	✓ <sup>3)</sup>
	ModelDesk	–	✓ <sup>2)</sup>
	Automotive Simulation Models	✓	–
	MotionDesk	–	✓ <sup>2)</sup>
	MotionDesk Blockset	✓	–
	Flight Rec Data Merger	–	✓ <sup>3)</sup>
	Further products of RCP and HIL software package	–	✓ <sup>3)</sup>

<sup>1)</sup> The 32-bit version also supports 64-bit MATLAB versions if the 32-bit MATLAB version from the same MATLAB release is installed on your host PC in parallel. For details see the notes below this table.

<sup>2)</sup> The 32-bit version also supports 64-bit versions of MATLAB.

<sup>3)</sup> The product is independent of the MATLAB architecture (32-bit/64-bit).

**Further product-specific limitations**

**AutomationDesk, dSPACE Python Extensions** The 32-bit version of AutomationDesk and dSPACE Python Extensions also supports 64-bit MATLAB versions if the 32-bit MATLAB version from the same MATLAB release is installed on your host PC in parallel. Both MATLAB versions (32-bit and 64-bit) must be connected to the corresponding installation set in the dSPACE Installation Manager. The 64-bit version has to be specified as the preferred connection. In this case, the MATLAB Access Library (AutomationDesk) and the matlplib2 Python modules (Python Extensions) also support reading and writing MAT files of file format version > 5.0.

**Restricted MAT file support** The following products only support reading and writing MAT files of file format version 5.0. MAT files of this version can be created in MATLAB by using the `save` command with option `'-v6'`:

- ModelDesk 4.0 (Maneuver Editor, Road Generator)
- ControlDesk Next Generation (ControlDesk 5.3)
- AutomationDesk 4.1 (MATLAB Access Library) without installation of 32-bit MATLAB



The MATLAB Access Library supports reading and writing MAT files of file format version > 5.0 if a 32-bit MATLAB installation of the same MATLAB release is installed on your host PC in addition to 64-bit MATLAB. See description above.

- The matlplib2 Python modules without installation of 32-bit MATLAB



The matlplib2 Python modules support reading and writing MAT files of file format version > 5.0 if a 32-bit MATLAB installation of the same MATLAB release is installed on your host PC in addition to 64-bit MATLAB. See description above.


**dSPACE HIL API .NET** dSPACE HIL API .NET does not support 64-bit MATLAB versions.

**RTI-MP** The `rtimpdiag` command is not functional. This command is based on dSPACE HIL API .NET, which does not support 64-bit MATLAB versions.

**Parallel installation of 32-bit and 64-bit MATLAB versions** On a 64-bit operating system, you can install both a 32-bit and a 64-bit version of a particular MATLAB release, e.g., R2013b. However, MATLAB release versions share the same preference settings folder, which means that you have to set all architecture-dependent settings such as the MEX compiler settings each time you switch between different versions of one MATLAB release.

---

### Limitations for the 64-bit version of TargetLink

**Importing an A2L file** It is not possible to import an A2L file into a 64-bit version of TargetLink. However, you can use a workaround described in *Basics of Importing A2L Files* ( *TargetLink Data Dictionary A2L Import and Export*).

## Limitations for Windows 7

---

<b>Objective</b>	Some limitations apply when you use Windows 7 in combination with dSPACE software.
<b>MATLAB support</b>	For system requirements of MathWorks® software, refer to <a href="http://www.mathworks.com/support/sysreq/current_release">http://www.mathworks.com/support/sysreq/current_release</a> .
<b>Fast user switching not supported</b>	The dSPACE software does not support the fast user switching feature of Windows.
<b>Closing dSPACE software before PC shutdown</b>	The shutdown procedure of Windows operating systems might cause some required processes to be aborted although they are still being used by dSPACE software. To avoid data loss, it is recommended to terminate the dSPACE software manually before a PC shutdown is performed.
<b>User Account Control</b>	It is recommended to disable Windows' User Account Control (UAC) during the installation of dSPACE software. If you cannot disable it, you should note the following Windows behavior: If UAC is enabled, the setup programs run with 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**

The first time dSPACE USB devices using cables with opto-isolation are connected to the PC, there might be a message that the device driver software was not successfully installed. The dSPACE device will nevertheless work properly later on.





**Numerics**

64-bit dSPACE DVD  
limitations 227

**A**

adaptation  
  Simulink configuration parameters 209

ASM Base InCylinder Blockset  
  migrating 42

ASM Diesel Engine Blockset  
  migrating 46  
  new features 43

ASM Diesel Exhaust Blockset  
  new features 48

ASM Drivetrain Basic Blockset  
  migrating 50  
  new features 50

ASM Electric Components Blockset  
  new features 52

ASM Engine Gasoline Basic Blockset  
  migrating 54  
  new features 53

ASM Engine Gasoline Blockset  
  migrating 59  
  new features 57

ASM Gasoline InCylinder Blockset  
  migrating 62

ASM Parameterization tool  
  new features 63

ASM Parameterization Tool  
  migrating 63

ASM Pneumatics Blockset  
  migrating 65  
  new features 65

ASM Traffic Blockset  
  migrating 66  
  new features 66

ASM Turbocharger Blockset  
  new features 67

ASM Vehicle Dynamics Blockset  
  migrating 70  
  new features 68

AutomationDesk  
  using with current release 39

AUTOSAR  
  TargetLink related  
  migration 204

**C**

Common Program Data folder 12

CommonProgramDataFolder 12

ControlDesk Next Generation  
  migration 93  
  new features 84

**D**

Documents folder 12

DocumentsFolder 12

DS1006 GNU C/C++ compiler 121

dSPACE FlexRay Configuration Package  
  new features 107

dSPACE HIL API .NET  
  new features 97

dSPACE Python Extensions  
  new features 99

dSPACE XIL API  
  migration 101  
  new features 101

DSPACE\_CONFIG 16

DSPACE\_ROOT 16

DVD contents 17

**E**

ECU Interface Manager  
  migration 104  
  new features 103

environment variables  
  set via command prompt 16

**F**

Firmware Manager  
  new features 105

**G**

general enhancements and changes 15

**H**

host PC software  
  operating system 224

**K**

key features 25

**L**

Limitations  
  TargetLink  
  obsolete limitations 214

limitations for products on 64-bit dSPACE  
  DVD 227

limitations for Windows 64-Bit and  
  dSPACE 64-Bit software 227

limitations for Windows 7 230

Local Program Data folder 12

LocalProgramDataFolder 12

**M**

MATLAB  
  supported releases 222

MicroAutoBox  
  new features 121

MicroLabBox  
  new features 119

Microtec PowerPC C/C++ compiler 122  
  migrating  
  ASM Base InCylinder Blockset 42  
  ASM Diesel Engine Blockset 46  
  ASM Drivetrain Basic Blockset 50

ASM Engine Gasoline Basic Blockset 54

ASM Engine Gasoline Blockset 59

ASM Gasoline InCylinder Blockset 62

ASM Parameterization Tool 63

ASM Pneumatics Blockset 65

ASM Traffic Blockset 66

ASM Vehicle Dynamics Blockset 70

migration  
  ControlDesk Next Generation 93

dSPACE XIL API 101

ECU Interface Manager 104

Model Compare 110

MotionDesk 116  
  RTI 122

RTI Bypass Blockset 127

RTI CAN MultiMessage Blockset 132

RTI FPGA Programming Blockset 141

RTI LIN MultiMessage Blockset 145

Model Compare  
  migration 110  
  new features 109

ModelDesk  
  new features 113

MotionDesk  
  migration 116  
  new features 115

**N**

new features  
  ASM Diesel Engine Blockset 43

  ASM Diesel Exhaust Blockset 48

  ASM Drivetrain Basic Blockset 50

  ASM Electric Components Blockset 52

  ASM Engine Gasoline Basic Blockset 53

  ASM Engine Gasoline Blockset 57

  ASM Parameterization tool 63

  ASM Pneumatics Blockset 65

  ASM Traffic Blockset 66

  ASM Turbocharger Blockset 67

  ASM Vehicle Dynamics Blockset 68

  ControlDesk Next Generation 84

  dSPACE FlexRay Configuration  
  Package 107

  dSPACE HIL API .NET 97

  dSPACE Python Extensions 99

  dSPACE XIL API 101

  ECU Interface Manager 103

  Firmware Manager 105

  MicroAutoBox 121

  MicroLabBox 119

  Model Compare 109

  ModelDesk 113

  MotionDesk 115

  Real-Time Testing 117

  RTI Bypass Blockset 125

  RTI CAN Blockset 129

  RTI CAN MultiMessage Blockset 131

  RTI Electric Motor Control Blockset 135

  RTI Ethernet Blockset 137

  RTI FPGA Programming Blockset 140

  RTI LIN MultiMessage Blockset 145

- RTI USB Flight Recorder Blockset 147
- RTI/RTI-MP 119
- RTLib 119
- SCALEXIO firmware 149
- SystemDesk 152
- VEOS 219
- new hardware 16
- not supported MATLAB features (R2014a) 76, 121
- not supported MATLAB features (R2014b) 77, 121

**P**

- product overview 22

**R**

- RCP and HIL software
  - definition 17
- Real-Time Testing
  - new features 117
- requirements
  - host PC software
    - operating system 224
- RTI Bypass Blockset
  - migration 127
  - new features 125
- RTI CAN Blockset
  - new features 129
- RTI CAN MultiMessage Blockset
  - migration 132
  - new features 131
- RTI Electric Motor Control Blockset
  - new features 135
- RTI Ethernet Blockset
  - new features 137
- RTI FPGA Programming Blockset
  - migration 141
  - new features 140
- RTI LIN MultiMessage Blockset
  - migration 145
  - new features 145
- RTI USB Flight Recorder Blockset
  - new features 147
- RTI/RTI-MP
  - new features 119
- RTLib
  - new features 119

**S**

- SCALEXIO firmware
  - new features 149
- Simulink
  - configuration parameters
    - adaptation 209
- supported MATLAB releases 222
- system requirements
  - operating system 224
- SystemDesk
  - new features 152

**T**

- TargetLink
  - API commands
    - changes 212
  - AUTOSAR features, new
    - supported releases 172
  - code changes
    - migration 189
  - code efficiency, improvement of 166
  - code generator options
    - backward compatibility 200
    - changed default value 200
  - discontinued features 213
  - migrating to
    - new version 184
  - migration
    - AUTOSAR related 204
    - code changes 189
    - obsolete limitations 214
    - various aspects 208
  - new API functions 181
  - new code generator options 177
  - new features 158
    - general changes 180
    - general enhancements 180
  - new version
    - migrating to 184
  - newly supported Simulink blocks 159
  - target support
    - discontinued compiler versions 175
    - discontinued evaluation boards 175
    - new compiler versions 175
    - new evaluation boards 175
    - supported targets 175
- TargetLink Data Dictionary
  - API commands
    - changes 212
    - new commands 170
  - discontinued features 213
  - migrating to
    - new version 184
  - migration 185
    - discontinued documentation 185
    - upgrading existing data dictionaries 186
  - new features 158
  - new version
    - migrating to 184

- limitations 230

**U**

- using AutomationDesk 4.1 39

**V**

- VEOS
  - new features 219
- version history 22

**W**

- Windows 64-Bit
  - limitations 227
- Windows 7