New Features and Migration

Release 7.4 – November 2012
How to Contact dSPACE

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

How to Contact dSPACE Support

There are different ways to contact dSPACE Support:
• Visit our website at http://www.dspace.com/goto?support
• Send an e-mail or phone:
  • General Technical Support:
support@dspace.de
    +49 5251 1638-941
  • TargetLink Support:
support.tl@dspace.de
    +49 5251 1638-700
• Use the dSPACE Installation Manager:
  • On your dSPACE DVD at \Tools\InstallationManager
  • Via Start – Programs – dSPACE Installation Manager (after installation of the dSPACE software)
  • At http://www.dspace.com/goto?im
You can always find the latest version of the dSPACE Installation Manager here.
dSPACE recommends that you use the dSPACE Installation Manager to contact dSPACE 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/goto?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 - 2012 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

This publication and the contents hereof are subject to change without notice.
AutomationDesk, CalDesk, ConfigurationDesk, ControlDesk, SCALEXIO, SYNECT, SystemDesk, TargetLink and VEOS are registered trademarks of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations.
# Contents

## About This Document

- Document Symbols and Conventions ........................................ 10
- Accessing Online Help and PDF Files ..................................... 11

## Overview of dSPACE Release 7.4

- General Enhancements and Changes ........................................ 14
- Product Version Overview ..................................................... 16
- New Product Key Features ..................................................... 19
- Migrating to dSPACE Release 7.4 .......................................... 25

## AutomationDesk

- New Features of AutomationDesk 3.6 ..................................... 27
- Migrating to AutomationDesk 3.6 .......................................... 30

## Automotive Simulation Models (ASM)

- All ASM Blocksets .................................................................... 32
  - New Features of All ASM Blocksets .................................... 32
- ASM Diesel Engine Blockset .................................................... 33
  - New Features of ASM Diesel Engine Blockset 1.6.1 ............ 33
  - Changes in the ASM Diesel Engine Demo Model ................. 33
  - Migrating to ASM Diesel Engine Blockset 1.6.1 .................. 34
- ASM Diesel Exhaust Blockset .................................................. 35
  - Migrating to ASM Diesel Exhaust Blockset 1.2.3 ............... 35
- ASM Drivetrain Basic Blockset ............................................... 35
  - Migrating to ASM Drivetrain Basic Blockset 1.6.1 ............ 35
- ASM Electric Components Blockset ......................................... 36
  - Migrating to ASM Electric Components Blockset 2.3 ......... 36
- ASM Environment Blockset ................................................... 37
  - New Features of ASM Environment Blockset 2.0.1 ............ 37
  - Migrating to ASM Environment Blockset 2.0.1 .................. 37
- ASM Engine Gasoline Basic Blockset .................................... 38
  - New Features of ASM Engine Gasoline Basic Blockset 1.4.1 ... 38
  - Changes in the ASM Engine Gasoline Basic Demo Model ....... 38
  - Migrating to ASM Engine Gasoline Basic Blockset 1.4.1 ....... 39
ASM Engine Gasoline Blockset ........................................................ 40
  New Features of ASM Engine Gasoline Blockset 2.4.1 ............... 40
  Changes in the ASM Engine Gasoline Demo Model .................. 40
  Migrating to ASM Engine Gasoline Blockset 2.4.1 .................... 41
ASM Optimizer .............................................................................. 42
  Migrating to ASM Optimizer Blockset 1.3.4 ............................. 42
ASM Traffic Blockset ................................................................. 42
  Migrating to ASM Traffic Blockset 1.3.1 ................................. 42
ASM Trailer Blockset .................................................................. 43
  New Features of ASM Trailer Blockset 2.0 .............................. 43
  Migrating to ASM Trailer Blockset 2.0 ................................. 43
ASM Truck Blockset .................................................................... 45
  Changes in the ASM Truck Demo Model ................................. 45
  Migrating to ASM Truck Blockset 1.4 .................................. 45
ASM Turbocharger Blockset ......................................................... 46
  Migrating to ASM Turbocharger Blockset 2.1 .......................... 46
ASM Vehicle Dynamics Blockset ................................................ 47
  New Features of ASM Vehicle Dynamics Blockset 2.3 ........... 47
  Migrating to ASM Vehicle Dynamics Blockset 2.3 .................. 47
ConfigurationDesk ...................................................................... 49
  ConfigurationDesk – Implementation ...................................... 49
  New Features of ConfigurationDesk 4.3 ................................. 49
  Migrating Projects from ConfigurationDesk 4.2 (on dSPACE Release 7.3) or Earlier .................................................... 52
Container Manager ..................................................................... 55
  New Features of the Container Manager 3.2 ........................... 55
ControlDesk 3.x ......................................................................... 57
  New Features of ControlDesk 3.7.4 .................................... 58
  Migrating to ControlDesk 3.7.4 ....................................... 58
ControlDesk Next Generation ....................................................... 59
  New Features of ControlDesk Next Generation (ControlDesk 4.3) .... 60
  New General Features (ControlDesk 4.3) .............................. 60
  New Features of Platform Management and Platforms/Devices (ControlDesk 4.3) ................................................. 62
  New Variable Management Features (ControlDesk 4.3) ......... 64
  New Instrument Features (ControlDesk 4.3) ........................... 65
  New Measurement and Recording Features (ControlDesk 4.3) 70
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Bus Navigator Features (ControlDesk 4.3)</td>
<td>70</td>
</tr>
<tr>
<td>New Data Set Management Features (ControlDesk 4.3)</td>
<td>70</td>
</tr>
<tr>
<td>New ECU Diagnostics Features (ControlDesk 4.3)</td>
<td>72</td>
</tr>
<tr>
<td>New Automation Features (ControlDesk 4.3)</td>
<td>73</td>
</tr>
<tr>
<td>Migrating to ControlDesk Next Generation (ControlDesk 4.3)</td>
<td>75</td>
</tr>
<tr>
<td>dSPACE HIL API .NET</td>
<td>77</td>
</tr>
<tr>
<td>New Features of dSPACE HIL API .NET 1.3</td>
<td>77</td>
</tr>
<tr>
<td>dSPACE FlexRay Configuration Package</td>
<td>79</td>
</tr>
<tr>
<td>New Features of dSPACE FlexRay Configuration Package 3.0</td>
<td>79</td>
</tr>
<tr>
<td>Migrating to dSPACE FlexRay Configuration Package 3.0</td>
<td>82</td>
</tr>
<tr>
<td>dSPACE Python Extensions</td>
<td>83</td>
</tr>
<tr>
<td>New Features of dSPACE Python Extensions 1.3</td>
<td>83</td>
</tr>
<tr>
<td>Migrating to dSPACE Python Extensions 1.3</td>
<td>84</td>
</tr>
<tr>
<td>ECU Interface Manager</td>
<td>85</td>
</tr>
<tr>
<td>New Features of ECU Interface Manager 1.2</td>
<td>85</td>
</tr>
<tr>
<td>Migrating to ECU Interface Manager 1.2</td>
<td>87</td>
</tr>
<tr>
<td>Model Compare</td>
<td>89</td>
</tr>
<tr>
<td>New Features of Model Compare 2.3</td>
<td>89</td>
</tr>
<tr>
<td>Migration to Model Compare 2.3</td>
<td>90</td>
</tr>
<tr>
<td>ModelDesk</td>
<td>91</td>
</tr>
<tr>
<td>New Features of ModelDesk 2.6</td>
<td>91</td>
</tr>
<tr>
<td>MotionDesk</td>
<td>93</td>
</tr>
<tr>
<td>New Features of MotionDesk 3.1</td>
<td>94</td>
</tr>
<tr>
<td>Migrating to MotionDesk 3.1</td>
<td>95</td>
</tr>
<tr>
<td>RapidPro Control Unit RTI Blockset</td>
<td>97</td>
</tr>
<tr>
<td>New Features of the RapidPro Control Unit RTI Blockset</td>
<td>97</td>
</tr>
<tr>
<td>Real-Time Testing</td>
<td>99</td>
</tr>
<tr>
<td>New Features of Real-Time Testing 2.0</td>
<td>99</td>
</tr>
<tr>
<td>Migrating to Real-Time Testing 2.0</td>
<td>99</td>
</tr>
</tbody>
</table>
Contents

RTI/RTI-MP and RTLib ............................................. 101
  New Features of RTI/RTI-MP and RTLib .......................... 101
  Migration Aspects of RTI/RTI-MP and RTLib ................... 103

RTI Bypass Blockset ............................................. 105
  New Features of the RTI Bypass Blockset 2.9.1 ................. 105
  Migrating to RTI Bypass Blockset 2.9.1 ......................... 107

RTI CAN MultiMessage Blockset .................................. 109
  New Features of the RTI CAN MultiMessage Blockset 2.7 ..... 109
  Migrating to RTI CAN MultiMessage Blockset 2.7 ............... 110

RTI LIN MultiMessage Blockset .................................. 111
  New Features of the RTI LIN MultiMessage Blockset 2.0 ...... 111
  Migrating to RTI LIN MultiMessage Blockset 2.0 ............... 112

RTI FPGA Programming Blockset .................................. 113
  New Features of the RTI FPGA Programming Blockset 2.4 .... 113
  Migrating to RTI FPGA Programming Blockset 2.4 ............... 115

SystemDesk 3.x .................................................. 117
  New Features of SystemDesk 3.2 ................................ 118
  New General Features............................................ 118
  Configuring and Generating V-ECUs ............................... 119
  Simulating Systems .............................................. 126
  Migrating to SystemDesk 3.2 ...................................... 127
    Migrating to SystemDesk 3.2 .................................... 127

SystemDesk 4.x .................................................. 129
  New Features of SystemDesk 4.0 ................................ 129
  Migrating to SystemDesk 4.0 ....................................... 130

TargetLink ...................................................... 131
  New Features of TargetLink 3.4 and TargetLink Data Dictionary 3.4 132
  New Production Code Generation Features ........................ 132
    Component-Based Development Using Abstract Interfaces... 133
    Improvement for Custom Code ................................ 134
    Optional Deactivation of Compute-Through-Overflow........... 134
    Generating Virtual ECUs for Virtual ECU Testing .......... 135
    Specifications of the Target Simulation Modules .......... 136
    Code Generator Options ...................................... 138
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>New API Commands</td>
<td>142</td>
</tr>
<tr>
<td>New Hook Functions</td>
<td>142</td>
</tr>
<tr>
<td>Access Function Changes</td>
<td>143</td>
</tr>
<tr>
<td>Improvement of Code Efficiency</td>
<td>144</td>
</tr>
<tr>
<td>General Enhancements and Changes</td>
<td>145</td>
</tr>
<tr>
<td>New AUTOSAR-Related Features</td>
<td>146</td>
</tr>
<tr>
<td>Features of the TargetLink AUTOSAR Module</td>
<td>146</td>
</tr>
<tr>
<td>New TargetLink Data Dictionary Features</td>
<td>150</td>
</tr>
<tr>
<td>Compare and Merge Features</td>
<td>151</td>
</tr>
<tr>
<td>Improvements for Data Dictionary File Handling</td>
<td>152</td>
</tr>
<tr>
<td>Further Improvements of the Data Dictionary Manager</td>
<td>155</td>
</tr>
<tr>
<td>New DD MATLAB API Commands</td>
<td>155</td>
</tr>
<tr>
<td>Migrating to TargetLink 3.4 and TargetLink Data Dictionary 3.4</td>
<td>158</td>
</tr>
<tr>
<td>Discontinued Data Dictionary Features</td>
<td>160</td>
</tr>
<tr>
<td>Migrating to TargetLink Data Dictionary 3.4</td>
<td>160</td>
</tr>
<tr>
<td>How to Upgrade a Data Dictionary Without Include Files</td>
<td>164</td>
</tr>
<tr>
<td>How to Upgrade a Data Dictionary With Include Files</td>
<td>165</td>
</tr>
<tr>
<td>How to Upgrade Libraries and Models from TargetLink 2.x to 3.x</td>
<td>167</td>
</tr>
<tr>
<td>Code Changes</td>
<td>168</td>
</tr>
<tr>
<td>Migration Aspects Regarding Code Generator Options</td>
<td>174</td>
</tr>
<tr>
<td>Changes in TargetLink and TargetLink Data Dictionary API</td>
<td>175</td>
</tr>
<tr>
<td>Functions</td>
<td>175</td>
</tr>
<tr>
<td>Various Migration Aspects</td>
<td>176</td>
</tr>
<tr>
<td>AUTOSAR-Related Migration Aspects</td>
<td>176</td>
</tr>
<tr>
<td>Obsolete Limitations</td>
<td>177</td>
</tr>
<tr>
<td>VEOS</td>
<td>179</td>
</tr>
<tr>
<td>New Features of VEOS 3.0</td>
<td>179</td>
</tr>
</tbody>
</table>

**Compatibility Information**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported MATLAB Releases</td>
<td>183</td>
</tr>
<tr>
<td>Operating System</td>
<td>184</td>
</tr>
<tr>
<td>Run-Time-Compatibility of dSPACE Software</td>
<td>187</td>
</tr>
<tr>
<td>Limitations for Windows Vista/Windows 7</td>
<td>188</td>
</tr>
<tr>
<td>Limitations for 64-Bit Windows Operating Systems in Combination with dSPACE 32-Bit Software</td>
<td>189</td>
</tr>
</tbody>
</table>

**Index**

191
About This Document

Contents
This document informs you about the new features of all the dSPACE software products in Release 7.4. 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

<table>
<thead>
<tr>
<th>Document Symbols and Conventions</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessing Online Help and PDF Files</td>
<td>11</td>
</tr>
</tbody>
</table>
Document Symbols and Conventions

Symbols

The following symbols may be used in this document.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![exclamation mark]</td>
<td>Indicates a general hazard that may cause personal injury of any kind if you do not avoid it by following the instructions given.</td>
</tr>
<tr>
<td>![lightning bolt]</td>
<td>Indicates the danger of electric shock which may cause death or serious injury if you do not avoid it by following the instructions given.</td>
</tr>
<tr>
<td>![triangle]</td>
<td>Indicates a hazard that may cause material damage if you do not avoid it by following the instructions given.</td>
</tr>
<tr>
<td>![information icon]</td>
<td>Indicates important information that should be kept in mind, for example, to avoid malfunctions.</td>
</tr>
<tr>
<td>![light bulb]</td>
<td>Indicates tips containing useful information to make your work easier.</td>
</tr>
</tbody>
</table>

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, for example, %DSPACE_PYTHON25% is the folder containing the Python installation.
- **<>** 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.
- **넷** Indicates that a link refers to another document, which is available in dSPACE HelpDesk.
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
Click Start – 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. Context-sensitive help is not available in all software products.

Help menu in the dSPACE software
From the menu bar, choose 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 7.4

Objective
Gives you an overview of the new key features in Release 7.4, and also information about unchanged products and general instructions on migrating.

Where to go from here
Information in this section

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Enhancements and Changes</td>
<td>14</td>
</tr>
<tr>
<td>Product Version Overview</td>
<td>16</td>
</tr>
<tr>
<td>New Product Key Features</td>
<td>19</td>
</tr>
<tr>
<td>Migrating to dSPACE Release 7.4</td>
<td>25</td>
</tr>
</tbody>
</table>
General Enhancements and Changes

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

Improving dSPACE software installation
As of dSPACE Release 7.0, improvements are being made to the installation programs for dSPACE software. The software architecture has been redesigned to reduce dependencies between products.

The aim is to ensure that software installations have maximal flexibility in installation and maximal run-time compatibility. This means that dSPACE products can be used in parallel after software installation, even if they are installed in different folders.

dSPACE Release 7.4 is an intermediate step towards this aim. Some products do not yet support all the new features.

For details, refer to Important Changes and Improvements to dSPACE Setup (Software Installation and Management Guide).

Restrictions when working with dSPACE HelpDesk
dSPACE HelpDesk is installed in release-specific folders in C:\Program Files\Common Files\dSPACE. For example, if you have installed products from dSPACE Release 7.4 and products from dSPACE Release 7.3, there are two dSPACE HelpDesks available.

(RCP and HIL software still accesses dSPACE HelpDesk in %DSPACE_ROOT%\Doc.)

Note the following restrictions:

If links to documents do not work, but return the error message Selection is not associated with any topics., the possible reasons are:

- The documents for the product are not installed anywhere 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 on the current dSPACE Release is unchanged, its user documentation is installed in the dSPACE HelpDesk version that the product setup was created for.

Each setup on dSPACE Release 7.4 installs its user documentation in dSPACE HelpDesk 7.4.

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

The printed user documentation is not delivered with Release 7.4 if you receive the release as an update for your existing dSPACE release. Use the current online help or PDF files to obtain information about new features, enhancements, and the current safety precautions regarding your products.

### Restriction for using Microsoft .NET Framework 4.5 Redistributables

The software products on dSPACE Release 7.4 do not support the Microsoft .NET Framework 4.5 Redistributables.

To avoid problems when using dSPACE software:

- If .NET Framework 4.5 Redistributables is installed on your host PC, you must uninstall it before you install the dSPACE software.
- Do not install the framework after installing dSPACE software.
# 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.

<table>
<thead>
<tr>
<th>Product</th>
<th>dSPACE Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.1</td>
</tr>
<tr>
<td>AutomationDesk</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Automotive Simulation Models</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ConfigurationDesk</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Container Manager</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ControlDesk 3.x&lt;sup&gt;1)&lt;/sup&gt;</td>
<td>3.7.1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ControlDesk Next Generation&lt;sup&gt;2)&lt;/sup&gt;</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>DCI Configuration Tool</td>
<td>2.4</td>
</tr>
<tr>
<td>dSPACE CAN API</td>
<td>2.5.1</td>
</tr>
<tr>
<td>dSPACE ECU Flash Programming Tool</td>
<td>2.2.1</td>
</tr>
<tr>
<td>dSPACE FlexRay Configuration Package</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>dSPACE HIL API .NET</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Product</td>
<td>dSPACE Release</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td>7.1</td>
</tr>
<tr>
<td>dSPACE Python Extensions</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ECU Interface Manager</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Compare</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ModelDesk</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>MotionDesk</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>MotionDesk Blockset</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Real-Time Testing</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI</td>
<td>6.7</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI-MP</td>
<td>6.3</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI AUTOSAR Package</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI Bypass Blockset</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI CAN Blockset</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI CAN MultiMessage Blockset</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI LIN MultiMessage Blockset</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- See *dSPACE Python Extensions* on page 83.
- See *ECU Interface Manager* on page 85.
- See *Model Compare* on page 89.
- See *ModelDesk* on page 91.
- See *MotionDesk* on page 93.
- See *Real-Time Testing* on page 99.
- See *RTI/RTI-MP and RTLib* on page 101.
- See *RTI Bypass Blockset* on page 105.
- See *RTI CAN MultiMessage Blockset* on page 109.
- See *RTI LIN MultiMessage Blockset* on page 111.
## Overview of dSPACE Release 7.4

### New Features and Migration November 2012

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.

<table>
<thead>
<tr>
<th>Product</th>
<th>dSPACE Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.1</td>
</tr>
<tr>
<td>RTI RapidPro Control Unit Blockset</td>
<td>1.10</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI FPGA Programming Blockset</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI Ethernet (UDP) Blockset</td>
<td>1.1</td>
</tr>
<tr>
<td>SystemDesk 3.x</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SystemDesk 4.x</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>TargetLink</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>TargetLink Data Dictionary</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable Editor</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>VEOS5)</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) ControlDesk 3.x will be 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 ([ControlDesk Next Generation Migration Guide](#)).

2) ControlDesk Next Generation is the successor to ControlDesk 3.x and CalDesk.

3) As of dSPACE Release 7.4, the DCI Configuration Tool comes with the DCI-GSI Configuration Package setup.

4) Also available as patch 1 for ECU Interface Manager 1.0 since April 2012.

5) VEOS is the successor to the dSPACE Offline Simulator.
## 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.

<table>
<thead>
<tr>
<th>Information in this topic</th>
<th>AutomationDesk on page 19</th>
<th>Automotive Simulation Models (ASMs) on page 19</th>
<th>ConfigurationDesk on page 20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Container management on page 20</td>
<td>ControlDesk Next Generation on page 20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dSPACE FlexRay Configuration Package on page 20</td>
<td>ECU Interface Manager on page 21</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model Compare on page 21</td>
<td>ModelDesk on page 21</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MotionDesk on page 21</td>
<td>Python Extensions on page 22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Real-Time Testing on page 22</td>
<td>RTI, RTI-MP and RTLib on page 22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTI Bypass Blockset on page 22</td>
<td>RTI CAN MultiMessage Blockset on page 22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTI LIN MultiMessage Blockset on page 22</td>
<td>RapidPro Control Unit RTI Blockset on page 23</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTI FPGA Programming Blockset on page 23</td>
<td>SystemDesk on page 23</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TargetLink on page 23</td>
<td>VEOS on page 25</td>
<td></td>
</tr>
</tbody>
</table>

**AutomationDesk**

The new key features of AutomationDesk are:

- The new Test Builder library providing blocks for creating tests with implicit basic features such as exception handling, logging and result evaluation.
- Enhancements to the Framework Builder library, the ControlDesk NG Access library and the HIL API library.
- Enhancements to data object handling in the Python Editor, and additional sort methods in the Change Order dialog.
- Enhancements to the report handling and report layout.

For details on the new features, refer to *New Features of AutomationDesk 3.6* on page 27.

**Automotive Simulation Models (ASMs)**

For details on the new features, refer to *Automotive Simulation Models (ASM)* on page 31.
Overview of dSPACE Release 7.4

**ConfigurationDesk**
For details on the new features, refer to *ConfigurationDesk – Implementation* on page 49.

**Container management**
The new key features of container management are:
- Exporting V-ECU implementations in SystemDesk 3.2 and TargetLink
- Exchanging SWCs between SystemDesk 4.0 and TargetLink
For details on the new features, refer to *New Features of the Container Manager 3.2* on page 55.

**ControlDesk Next Generation**
The new key features of ControlDesk Next Generation (ControlDesk 4.3) are:
- Camera interface and Video Monitoring instrument
- Avionics instruments
- Improved scale-using instruments
- VEOS support: triggered measurements on VEOS platform
- Automation: reading and writing variables directly without having to use instruments
- Support of ASAM MDF 4.0 for measurements
- ECU Diagnostics: Working with multiple ODX databases
- Bus Navigator:
  - Enhancements when using PC-based interfaces
  - Automation support
- Combined variable filter offering more flexibility
For details on the new features, refer to *New Features of ControlDesk Next Generation (ControlDesk 4.3)* on page 60.

**dSPACE FlexRay Configuration Package**
The new key features of the dSPACE FlexRay Configuration Package are:
- Support of PDU transmission modes
- Update bit behavior according to AUTOSAR
- Support of static event PDUs
The new key features of the dSPACE FlexRay Configuration Tool are:
- Support of AUTOSAR System Template 3.2.2
- Alias name for physical bus access
The new key feature of the FlexRay Configuration Blockset is:
- Assigning tasks to timetables via Runnable Function blocks of the Model Port Block Library

For details on the new features, refer to *New Features of dSPACE FlexRay Configuration Package 3.0* on page 79.

**ECU Interface Manager**
The new key features of the ECU Interface Manager are:
- Automation interface
- Password protection for ECU applications
- Deactivating function calls and write accesses

For details on the ECU Interface Manager, refer to *New Features of ECU Interface Manager 1.2* on page 85.

**Model Compare**
The new key features of Model Compare are:
- Enhanced tree view display options
- Tree view display considering comments
- Highlighting with smarter window opening
- Savable highlighting
- Support of Simulink SLX model format

For details on the new features, refer to *New Features of Model Compare 2.3* on page 89.

**ModelDesk**
The new key features of ModelDesk are:
- Support of VEOS
- Support of Simulink SLX model format
- Extended tool automation for automated handling of parameter files and variants

For details on the new features, refer to *New Features of ModelDesk 2.6* on page 91.

**MotionDesk**
The new key features of MotionDesk are:
- Automation interface
- Visualization of rain and snow

For details on the new features, refer to *New Features of MotionDesk 3.1* on page 94.
### Python Extensions
The new key feature of Python Extensions is:
- New API for platform management

For details on the new features, refer to *dSPACE Python Extensions* on page 83.

### Real-Time Testing
The new key feature of Real-Time Testing is:
- Support of VEOS

For details on the new features, refer to *New Features of Real-Time Testing 2.0* on page 99.

### RTI, RTI-MP and RTLib
The new key features of RTI, RTI-MP and RTLib are:
- Support of Simulink SLX model format
- Enhancements to the implementation software for MicroAutoBox II
- New procedure for loading slave applications to a DS2302 board

For details on the new features, refer to *New Features of RTI/RTI-MP and RTLib* on page 101.

### RTI Bypass Blockset
The new key features of the RTI Bypass Blockset are:
- Enhancements to the RTIBYPASS_FUNCTION_BLx block
- Reusing existing ECU variables with the internal bypass model
- Support of ASAM MCD-2 MC Ver. 1.6

The new key feature of the RTI Bypass Blockset MATLAB API is:
- Support of enhancements to the RTIBYPASS_FUNCTION_BLx block

For details on the new features, refer to *New Features of the RTI Bypass Blockset 2.9.1* on page 105.

### RTI CAN MultiMessage Blockset
The new key feature of the RTI CAN MultiMessage Blockset is:
- Support of AUTOSAR System Template 3.2.2

For details on the new features, refer to *New Features of the RTI CAN MultiMessage Blockset 2.7* on page 109.

### RTI LIN MultiMessage Blockset
The new key features of the RTI LIN MultiMessage Blockset are:
- Support of AUTOSAR System Template 3.2.2
- Support of LDF files according to the SAE J2602 standard

For details on the new features, refer to *New Features of the RTI LIN MultiMessage Blockset 2.0* on page 111.
### RapidPro Control Unit RTI Blockset

The new key features of the RapidPro Control Unit RTI Blockset are:
- Arbitrary crankshaft wheel design
- Detection of reverse crankshaft rotation
- Smoothed speed measurement

For details on the new features, refer to *New Features of the RapidPro Control Unit RTI Blockset* on page 97.

### RTI FPGA Programming Blockset

The new key features of the RTI FPGA Programming Blockset are:
- Extended Xilinx® software support
- Verilog support for handcoded FPGA applications
- Enhanced support of framework migration

For details on the new features, refer to *New Features of the RTI FPGA Programming Blockset 2.4* on page 113.

### SystemDesk

As of dSPACE Release 7.4, there are two different SystemDesk versions due to the conceptual differences between the AUTOSAR Releases 3.x and 4.x supported by SystemDesk:
- SystemDesk 3.x supports AUTOSAR 3.x.
- SystemDesk 4.0 supports AUTOSAR 4.0.2 and 4.0.3.

**SystemDesk 3.x**  
The new key features of SystemDesk 3.x are:
- Enhanced options for configuring and generating V-ECUs for test scenarios
- New simulation scenarios for V-ECU systems

For details on the new features, refer to *New General Features* on page 118.

**SystemDesk 4.x**  
The key features of SystemDesk 4.0 are:
- Complete support of AUTOSAR 4.0.2 and 4.0.3
- Interoperability with other AUTOSAR-compliant tools

For details on the features, refer to *New Features of SystemDesk 4.0* on page 129.

### TargetLink

The new key features of TargetLink are:
- Component-based development using abstract interfaces
- Custom code:
  - Enhancing unsupported Simulink blocks to TargetLink
  - Width-invariant custom code template
- Optional deactivation of compute-through-overflow
- Generating virtual ECUs for virtual ECU testing
- Target simulation modules:
  - New evaluation board: Renesas AB_050_Fx4
  - New compiler versions
- New code generator options
- New API commands
- New hook functions
- Access function changes
  - Specifying structure access functions
  - Propagation into structures
  - Controlling use of auxiliary variables
- Improved code efficiency
- Support of new Simulink SLX model format

For details on the new features, refer to *New Production Code Generation Features* on page 132.

The new key features of the TargetLink AUTOSAR module are:
- Support of AUTOSAR 4.03 and 3.22.
- Using component-based development
- Improved frame model generation including update functionality
- Delivery of software components as object code

For details on the new features, refer to *New AUTOSAR-Related Features* on page 146.

The new key features of the TargetLink Data Dictionary are:
- Improved visualization and handling for comparing/merging full DDs or subtrees
- Simplified loading and inclusion of partial Data Dictionary files
- Simplified partition of a DD project file
- Improved way to add properties to the Object Explorer
- New DD MATLAB API commands

For details on the new features, refer to *New TargetLink Data Dictionary Features* on page 150.

For details on the TargetLink migration aspects (TargetLink, TargetLink AUTOSAR module, TargetLink Data Dictionary), refer to *Migrating to TargetLink 3.4 and TargetLink Data Dictionary 3.4* on page 158.
VEOS is the successor to the dSPACE Offline Simulator. The new features described below are enhancements and changes compared with dSPACE Offline Simulator 2.2p2.

The new key features of VEOS are:
- New dSPACE Target for Offline Simulation features including Real-Time Testing support
- Improved support for stimulation scenarios
- New VEOS Player

For details on the new features, refer to VEOS on page 179.

Migrating to dSPACE Release 7.4

Objective
After you install Release 7.4, some additional steps may be necessary.

Migrating from dSPACE Release 7.3
There are no general migration steps to be done. Product-specific migration steps are usually done automatically by the products. For exceptions, refer to the product-specific migration descriptions.

Migrating from dSPACE Release 7.2 or earlier
To migrate from dSPACE Release 7.2 or earlier to Release 7.4, you also have to perform the migration steps of the intervening dSPACE Releases. All of the required migration steps can be done with Release 7.4 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 New Features and Migration documents for previous releases are available via Internet and on the dSPACE DVD:
- Read them from the dSPACE DVD (see the \Doc\Print\PreviousReleases folder). The PDF files are called NewFeaturesAndMigrationxx.pdf, where xx stands for the release number.
New Features of AutomationDesk 3.6

New automation library

Test Builder library  AutomationDesk now has a Test Builder library, which offers automation blocks comparable to the Test Framework library. In contrast to the Test Framework library, the Test Builder library is based on the Framework Builder library and therefore uses its slot concept. This gives you greater flexibility in implementing the internal features such as exception handling, logging and result evaluation. The Test Builder library will replace the Test Framework library in the future.

For further information, refer to Test Builder (AutomationDesk Library Reference).

Enhancements to the libraries

The following libraries have been enhanced:

ControlDesk NG Access  The ControlDesk NG Access library now provides two new automation blocks:

- ReadVariableValue
  This block is used to read the value of a specified variable.

- WriteVariableValue
  This block is used to write a value to a specified variable.
For further information, refer to ControlDesk NG Access (AutomationDesk Library Reference).

**HIL API library**  The HIL API library provides some new data objects:
- The new Symbol common data object for using symbols.
- The new Vendor data object for switching between vendor-specific HIL API implementations. The default is dSPACE. The Vendor data object is provided by the following automation blocks:
  - InitBaseValue
  - InitConditionWatcher
  - InitDurationWatcher
  - InitMAPort
- There are some new data objects for capture handling:
  - CaptureResultReader (replaces CaptureResultIDFReader)
  - CaptureResultWriter (replaces CaptureResultIDFWriter)
  - The Start automation block now contains the CaptureResultWriter data object.
- There are some new data objects for stimulus handling:
  - SignalSegment
  - SignalDescription
  - SignalDescriptionsReader
  - SignalDescriptionsWriter

For further information, refer to HIL API (AutomationDesk Library Reference).

**Framework Builder library**  The library now provides the SequenceFrame block for creating custom templates on sequence level.

For further information, refer to Framework Builder (AutomationDesk Library Reference).

**Enhancements to project handling**
- The Change Children Order dialog has a new option to sort data objects alphanumerically or by type. For example, this helps you organize variables and parameters more clearly.
  For further information, refer to Change Children Order (AutomationDesk Reference).
- If you drag a data object from the Project Manager to the Python Editor of an opened Exec block, the data object name is now automatically added to the script with its _AD_ alias.
Enhancements to report handling

- If you execute the Export Report command, the Save As dialog the file name now suggests a file name consisting of the parent’s element name and the result name. For example, if you want to export the result named Result5 from the sequence named MySequence, the file name suggested for the exported report is MySequence_Result5.

- For easier navigation in reports, more bookmarks (PDF) or tree nodes (HTML) are automatically added to each report. For example, each test step is now be represented by a bookmark or tree node.

Enhancements to platform management

Enhanced platform support

VEOS, the new offline simulation platform, is supported by AutomationDesk using the HIL API library or the Platform Access library.

DSPACE Release 7.4 provides VEOS as a new software product to perform offline simulation on your host PC, but you must install an add-on to dSPACE Release 7.4 to work with VEOS in connection with AutomationDesk 3.6. The add-on is available at http://www.dspace.com/go/VEOS-AddOn.

Enhanced platform management functions

AutomationDesk provides further commands for platform management:

- Clear Flash
- Clear System
- Create Support Info
- Display Platform Message Dialogs
- Update Firmware

For further information, refer to Platform Manager (AutomationDesk Reference).
Migrating to AutomationDesk 3.6

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

For further information, refer to Migrating AutomationDesk (AutomationDesk Guide).

Migration aspect when using HIL API library
With AutomationDesk 3.6, the data type of the TaskNames data object in the GetTaskNames automation block has changed from Dictionary to List. The data objects are automatically migrated if you load an AutomationDesk project created with an earlier version, but the references which you specified for parameterizing this data object remain unchanged.

- Use the Find Inconsistencies dialog to search for invalid references. The search result will contain the references to the TaskNames data objects to be migrated manually.

You must also do this for the OfflineTaskNames data object in the GetTaskNames automation block.
Automotive Simulation Models (ASM)

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All ASM Blocksets</strong></td>
<td>32</td>
</tr>
<tr>
<td>ASM Diesel Engine Blockset</td>
<td>33</td>
</tr>
<tr>
<td>ASM Diesel Exhaust Blockset</td>
<td>35</td>
</tr>
<tr>
<td>ASM Drivetrain Basic Blockset</td>
<td>35</td>
</tr>
<tr>
<td>ASM Electric Components Blockset</td>
<td>36</td>
</tr>
<tr>
<td>ASM Environment Blockset</td>
<td>37</td>
</tr>
<tr>
<td>ASM Engine Gasoline Basic Blockset</td>
<td>38</td>
</tr>
<tr>
<td>ASM Engine Gasoline Blockset</td>
<td>40</td>
</tr>
<tr>
<td>ASM Optimizer</td>
<td>42</td>
</tr>
<tr>
<td>ASM Traffic Blockset</td>
<td>42</td>
</tr>
<tr>
<td>ASM Trailer Blockset</td>
<td>43</td>
</tr>
<tr>
<td>ASM Truck Blockset</td>
<td>45</td>
</tr>
<tr>
<td>ASM Turbocharger Blockset</td>
<td>46</td>
</tr>
<tr>
<td>ASM Vehicle Dynamics Blockset</td>
<td>47</td>
</tr>
</tbody>
</table>

Information in other sections

|Migrating ASM Models (ASM User Guide)| Provides general information on the migration process of ASM models.|
## All ASM Blocksets

### New Features of All ASM Blocksets

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support of new Simulink SLX model format</td>
<td>ASM and ModelDesk now support the new Simulink SLX model format. With MATLAB R2012b, models containing ASM blocks can be opened or saved in MDL or SLX format. Both formats can be used in ModelDesk projects and for custom components libraries.</td>
</tr>
<tr>
<td>Problems using ASM with MATLAB R2012a and R2012b</td>
<td>Due to performance problems with in MATLAB R2012a and R2012b, it is recommended to install the following bugfix from the Mathworks® website before using ASM with MATLAB R2012a or 2012b: <a href="http://www.mathworks.de/support/search_results.html?q=827771">http://www.mathworks.de/support/search_results.html?q=827771</a></td>
</tr>
</tbody>
</table>
ASM Diesel Engine Blockset

Where to go from here

Information in this section

New Features of ASM Diesel Engine Blockset 1.6.1

Changes in the ASM Diesel Engine Demo Model

Migrating to ASM Diesel Engine Blockset 1.6.1

New Features of ASM Diesel Engine Blockset 1.6.1

MEASUREMENT_INTERFACE block

The ASM Diesel Engine models can now replicate testbench measurement data during real-time simulation to validate the actual simulation results in a specific operating point. This feature has been included in the ASM Parameterization demo projects and in the Simulink demo model. ControlDesk layouts are provided for validation.

Changes in the ASM Diesel Engine Demo Model

MDLUserInterface/Engine Diesel/MDL_MEAS

The structure of the measurement interface feature has been changed:
New subsystem MDL_MEAS.

MDLUserInterface/Environment/MDL_PAR

The structure of the support of the measurement interface feature has been changed. The definition of unit for the ambient temperature has been changed from [mbar] to [Pa] for consistency.

MDL/Environment/Maneuver

The structure of the support of the measurement interface feature has been changed.

MDL/Engine/Cooling System

The structure of the support of the measurement interface feature has been changed. Adaptation of Goto/From block tags respectively.

MDL/Environment/Road

The definition of unit for the ambient temperature has been changed from [mbar] to [Pa].

MDL/Engine/AirPath/Turbo

The input at the MAPS_TC block has been changed.
### Migrating to ASM Diesel Engine Blockset 1.6.1

<table>
<thead>
<tr>
<th>Block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXHAUST_MANIFOLD block</td>
<td>A reset condition has been inserted to switch between map-based and physical turbocharger approach during simulation.</td>
</tr>
<tr>
<td>SOFT_ECU_DIESEL block</td>
<td>The extrapolation method for injection timing and smoke limitation map has been changed to interpolation-extrapolation (for homogenous and stratified mode) due to ModelDesk conformity. The crank angle bus signal has been renamed to correct the angle unit.</td>
</tr>
<tr>
<td>SWITCHES_TURBO block</td>
<td>The link to former version has been moved due to ModelDesk conformity by using the MAPS_TC parameter page.</td>
</tr>
</tbody>
</table>
ASM Diesel Exhaust Blockset

Migrating to ASM Diesel Exhaust Blockset 1.2.3

<table>
<thead>
<tr>
<th>Block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCR_CATALYST</td>
<td>The line in subsystem name “SCR Catalyst \n for each cell” has been removed to comply with naming conventions.</td>
</tr>
</tbody>
</table>

ASM Drivetrain Basic Blockset

Migrating to ASM Drivetrain Basic Blockset 1.6.1

<table>
<thead>
<tr>
<th>Block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEARBOX_MT</td>
<td>To simplify the mode, the PT1 element with time constant 0.001 was deleted from the Trq_Trm_In[Nm] outport because it had negligible effect. A unit delay is added during migration.</td>
</tr>
<tr>
<td>GEARBOX_AT</td>
<td>To simplify the mode, the PT1 element with time constant 0.001 was deleted from the Trq_Trm_In[Nm] outport because it had negligible effect. A unit delay is added during migration.</td>
</tr>
</tbody>
</table>
## ASM Electric Components Blockset

### Migrating to ASM Electric Components Blockset 2.3

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMSM_CONTROLLER block</td>
<td>The decoupling equation of the current controller has been corrected.</td>
</tr>
<tr>
<td>PMSM_CONTROLLER_THREE_LEVEL block</td>
<td>The decoupling equation of the current controller has been corrected.</td>
</tr>
<tr>
<td>BATTERY block</td>
<td>The Goto-From blocks have been replaced with signal lines in the ASMSignalBus. This solves a problem with multiple instances of the battery block.</td>
</tr>
<tr>
<td>BATTERY_MULTICELL block</td>
<td>The Goto/From blocks have been replaced with signal lines in the ASMSignalBus. This solves a problem with multiple instances of the battery block.</td>
</tr>
</tbody>
</table>
ASM Environment Blockset

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>New Features of ASM Environment Blockset 2.0.1</th>
<th>37</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrating to ASM Environment Blockset 2.0.1</td>
<td>37</td>
</tr>
</tbody>
</table>

New Features of ASM Environment Blockset 2.0.1

Lane sensor

The lane sensor can now be parameterized using ModelDesk.

Migrating to ASM Environment Blockset 2.0.1

CONTROLLER block

Braking during slow acceleration after startup is now avoided.

MANEUVER_SCHEDULER block

During initialization of the manual driving state, a fixed value was used to initialize the vehicle height. This has been replaced by a small offset to the tire radius.

ROAD block

The calculation of the length inside spline segments has been modified.

v_ROAD_REF block

Braking at maneuver segment transition between two consecutive 'FinalVelocity' segments is now avoided.
ASM Engine Gasoline Basic Blockset

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of ASM Engine Gasoline Basic Blockset 1.4.1</td>
<td>38</td>
</tr>
<tr>
<td>Changes in the ASM Engine Gasoline Basic Demo Model</td>
<td>38</td>
</tr>
<tr>
<td>Migrating to ASM Engine Gasoline Basic Blockset 1.4.1</td>
<td>39</td>
</tr>
</tbody>
</table>

New Features of ASM Engine Gasoline Basic Blockset 1.4.1

The ASM Gasoline Engine Basic models can now replicate testbench measurement data during real-time simulation to validate simulation results at a specific operating point. This feature has been included in the ASM Parameterization demo projects and the Simulink demo model. ControlDesk layouts are provided for validation.

Changes in the ASM Engine Gasoline Basic Demo Model

<table>
<thead>
<tr>
<th>MDLUserInterface/Engine GasBas/MDL_MEAS</th>
<th>The structure of the measurement interface feature has been changed: New subsystem MDL_MEAS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDLUserInterface/Environment/MDL_PAR</td>
<td>The structure of the support of the measurement interface feature has been changed. The definition of unit for the ambient temperature has been changed from [mbar] to [Pa] for consistency.</td>
</tr>
<tr>
<td>MDL/Environment/Maneuver</td>
<td>The structure of the support of the measurement interface feature has been changed.</td>
</tr>
<tr>
<td>MDL/Engine/Cooling System</td>
<td>The structure of the support of the measurement interface feature has been changed: Adaptation of Goto/From block tags respectively.</td>
</tr>
<tr>
<td>MDL/Environment/Road</td>
<td>The unit defined for the ambient temperature has changed from [mbar] to [Pa].</td>
</tr>
</tbody>
</table>
Automotive Simulation Models (ASM)

MDL/Engine/AirPath/Turbo

The input at the MAPS_TC block has been changed.

Migrating to ASM Engine Gasoline Basic Blockset 1.4.1

WALL_FILM block

The extrapolation method for the time delay map has been changed to interpolation-extrapolation due to ModelDesk conformity.
ASM Engine Gasoline Blockset

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of ASM Engine Gasoline Blockset 2.4.1</td>
<td>40</td>
</tr>
<tr>
<td>Changes in the ASM Engine Gasoline Demo Model</td>
<td>40</td>
</tr>
<tr>
<td>Migrating to ASM Engine Gasoline Blockset 2.4.1</td>
<td>41</td>
</tr>
</tbody>
</table>

New Features of ASM Engine Gasoline Blockset 2.4.1

MEASUREMENT_INTERFACE block

The ASM Gasoline Engine Basic models can now replicate testbench measurement data during real-time simulation to validate simulation results at a specific operating point. This feature has been included in the ASM Parameterization demo projects and the Simulink demo model. ControlDesk layouts are provided for validation.

Changes in the ASM Engine Gasoline Demo Model

MDLUserInterface/Engine Gasoline/MDL_MEAS

The structure of the measurement interface feature has been changed:
New subsystem MDL_MEAS.

MDLUserInterface/Environment/MDL_PAR

The structure of the support of the measurement interface feature has been changed. The definition of unit for the ambient temperature has been changed from [mbar] to [Pa] for consistency.

MDL/Environment/Maneuver

The structure of the support of the measurement interface feature has been changed.

MDL/Engine/Cooling System

The structure of the support of the measurement interface feature has been changed: Adaptation of Goto/From block tags.

MDL/Environment/Road

The definition of unit for the ambient temperature has been changed from [mbar] to [Pa].

MDL/Engine/AirPath/Turbo

The input at the MAPS_TC block has been changed.
# Migrating to ASM Engine Gasoline Blockset 2.4.1

<table>
<thead>
<tr>
<th>Block Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXHAUST_MANIFOLD block</td>
<td>A reset condition has been inserted to switch between the map-based and physical turbocharger approach during simulation.</td>
</tr>
<tr>
<td>SOFT_ECU_DIESEL block</td>
<td>The extrapolation method for injection timing and smoke limitation map has been changed to interpolation-extrapolation (for homogenous and stratified mode) due to ModelDesk conformity.</td>
</tr>
<tr>
<td>WALL_FILM block</td>
<td>The extrapolation method for the time delay map has been changed to interpolation-extrapolation due to ModelDesk conformity.</td>
</tr>
<tr>
<td>SWITCHES_TURBO block</td>
<td>The link to the former version has been moved due to ModelDesk conformity by using the MAPS_TC parameter page.</td>
</tr>
</tbody>
</table>

**Related topics**
- Basics
  - Migrating ASM Models (`ASM User Guide`)
ASM Optimizer

Migrating to ASM Optimizer Blockset 1.3.4

| Loading results of previous optimization tasks | When loading the results of a previous optimization task, it was only assigned to the last parameter in the parameter list. Further, the assigned value was rounded by mistake. These issues were solved. |

ASM Traffic Blockset

Migrating to ASM Traffic Blockset 1.3.1

<table>
<thead>
<tr>
<th>FELLOW_PARAMETERS block</th>
<th>Zero lengths of fellow vehicles are now avoided to prevent a division by zero.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FELLOW_POSITION block</td>
<td>The step size of a discrete integrator is now a block parameter.</td>
</tr>
<tr>
<td>TRAFFIC_SCHEDULER block</td>
<td>Traffic scenarios with continue segments after initial segments resulted in inconsistent simulation results. This has been corrected.</td>
</tr>
</tbody>
</table>
ASM Trailer Blockset

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of ASM Trailer Blockset 2.0</td>
</tr>
<tr>
<td>Migrating to ASM Trailer Blockset 2.0</td>
</tr>
</tbody>
</table>

New Features of ASM Trailer Blockset 2.0

Dolly extension

The ASM Trailer blockset now supports simulating a dolly extension, see the following illustration.

A new demo model of ASM Truck shows the simulation with a semi-trailer or dolly extension.

Migrating to ASM Trailer Blockset 2.0

<table>
<thead>
<tr>
<th>TIRE_MODEL_MAGIC_FORMULA_&lt;xyz&gt; block</th>
<th>Two new inports have been added to the blocks for the tire radius to be changed by an external signal.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIRE_MODEL_TMEASY_&lt;xyz&gt; block</td>
<td>Two new inports have been added to the blocks for the tire radius to be changed by an external signal. The scaling parameters are connected to the slip camber and overturning torque calculation.</td>
</tr>
<tr>
<td>RELATIVE_VELOCITIES_TRAILER_&lt;xyz&gt; block</td>
<td>The Steering rod velocity inport is now a 2-dimensional signal to enable the left and right wheels to be steered independently.</td>
</tr>
<tr>
<td>Block Name and Parameters</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUSKIN_TRAILER_.&lt;xyz&gt;_.3DOF block</td>
<td>The <em>Steering rod displacement</em> and <em>Steering rod velocity</em> inports are now 2-dimensional signals to enable the left and right wheels to be steered independently.</td>
</tr>
<tr>
<td>BALL_JOINT_ACCELERATION_TRAILER block</td>
<td>This subsystem can now be switched to simulate a mechanism with 1 degree of freedom.</td>
</tr>
<tr>
<td>BALL_JOINT_ARTICULATED_BODY_TRAILER block</td>
<td>This subsystem can now be switched to simulate a mechanism with 1 degree of freedom.</td>
</tr>
<tr>
<td>HITCH_POS_TRAILER block</td>
<td>The block name has been changed from HITCH_POS_VEHICLE to HITCH_POS_TRAILER.</td>
</tr>
<tr>
<td>HITCH_VELOCITY_TRAILER block</td>
<td>The block name has been changed from HITCH_VELOCITY_VEHICLE to HITCH_VELOCITY_TRAILER.</td>
</tr>
<tr>
<td>CONTACT_POINT_CALCULATION_.&lt;xyz&gt; block</td>
<td>You can reset the tire radius calculation.</td>
</tr>
</tbody>
</table>
ASM Truck Blockset

Where to go from here

Information in this section

| Changes in the ASM Truck Demo Model | 45 |
| Migrating to ASM Truck Blockset 1.4 | 45 |

Changes in the ASM Truck Demo Model

ASM.TruckTrailer

The demo model can now be simulated with a dolly extension (see New Features of ASM Trailer Blockset 2.0 on page 43.

Migrating to ASM Truck Blockset 1.4

| TIRE_MODEL_MAGIC_.<xyz> block | Two new inports have been added to the blocks for the tire radius to be changed by an external signal. |
| TIRE_MODEL_TMEASY_.<xyz> block | Two new inports have been added to the blocks for the tire radius to be changed by an external signal. The scaling parameters are connected to the slip camber and overturning torque calculation. |
| RELATIVE_VELOCITIES_REAR_.<xyz> block | The Steering rod velocity inport is now a 2-dimensional signal to enable the left and right wheels to be steered independently. |
| SUSKIN_REAR_.<xyz>_.3DOF block | The Steering rod displacement and Steering rod velocity inports are now 2-dimensional signals to enable the left and right wheels to be steered independently. |
| CONTACT_POINT_CALCULATION_.<xyz> block | You can reset the tire radius calculation. |
ASM Turbocharger Blockset

Migrating to ASM Turbocharger Blockset 2.1

COMPRESSOR block

The COMPRESSOR block has been modified to reflect a certain behavior during instable surge operating condition. This operating condition is detected during simulation by comparing the current pressure ratio with the maximum stable pressure ratio. This mechanism requires information about the margin of the stable operating region, which is now contained in the surge line and the inverse surge line.

The parameterization files for this block have a new extrapolation option to simulate an appropriate pressure ratio during surge and backflow conditions.

COMPRESSOR_HP block

Refer to Compressor HP (ASM Turbocharger Reference).

MAPS_TC block

A link to the MAPS_TC block is changed to the former version MAPS_TC_6.0 block during migration to guarantee the same block behavior after migration.

The new MAPS_TC block contains several changes:

- The turbo control influence on efficiency has been modified from additive to multiplicative.
- The temperatures are evaluated relatively instead of absolutely to account for changes in the ambient conditions.
- The variable naming for compressor output pressure limits has been corrected.
ASM Vehicle Dynamics Blockset

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>New Features of ASM Vehicle Dynamics Blockset 2.3</th>
<th>47</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrating to ASM Vehicle Dynamics Blockset 2.3</td>
<td>47</td>
</tr>
</tbody>
</table>

New Features of ASM Vehicle Dynamics Blockset 2.3

Subframe

The elastokinematics of a subframe are now used in addition to the suspension compliances. The SUBFRAME block can be used in the model as an option and in parallel to the suspension compliance block.

Migrating to ASM Vehicle Dynamics Blockset 2.3

<table>
<thead>
<tr>
<th>TIRE_MODEL_MAGIC_.FORMULA_.&lt;xyz&gt; block</th>
<th>Two new inports have been added to the blocks for the tire radius to be changed by an external signal.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIRE_MODEL_TMEASY_.&lt;xyz&gt; block</td>
<td>Two new inports have been added to the blocks for the tire radius to be changed by an external signal. The scaling parameters are connected to the slip camber and overturning torque calculation.</td>
</tr>
<tr>
<td>RELATIVE_VELOCITIES_.&lt;xyz&gt; block</td>
<td>The Steering rod velocity import is now a 2-dimensional signal to enable the left and right wheels to be steered independently.</td>
</tr>
<tr>
<td>CONTACT_POINT_.CALCULATION_.&lt;xyz&gt; block</td>
<td>You can reset the tire radius calculation.</td>
</tr>
<tr>
<td>Steering left and right wheel</td>
<td>The Steering rod displacement and Steering rod velocity imports are now 2-dimensional signals to enable the left and right wheels to be steered independently. This applies to the following blocks:</td>
</tr>
<tr>
<td></td>
<td>• RIGID_AXLE</td>
</tr>
<tr>
<td></td>
<td>• SEMI_TRAILING_ARM</td>
</tr>
<tr>
<td></td>
<td>• MC_PHERSON_STRUT</td>
</tr>
<tr>
<td></td>
<td>• SUSKIN_REAR_SYM_3DOF</td>
</tr>
</tbody>
</table>
Automotive Simulation Models (ASM)

- SUSKIN_REAR_SYMMETRIC_3DOF
- SUSPENSION_KINEMATICS_REAR_SYMMETRIC
- SUSPENSION_KINEMATICS_REAR_ASYMMETRIC
- SUSKIN_FRONT_SYMMETRIC_3DOF
- SUSKIN_FRONT_ASYMMETRIC_3DOF
- SUSPENSION_KINEMATICS_FRONT_SYMMETRIC
- SUSPENSION_KINEMATICS_FRONT_ASYMMETRIC
ConfigurationDesk

ConfigurationDesk – Implementation

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of ConfigurationDesk 4.3</td>
<td>49</td>
</tr>
<tr>
<td>Migrating Projects from ConfigurationDesk 4.2 (on dSPACE Release 7.3) or Earlier</td>
<td>52</td>
</tr>
</tbody>
</table>

New Features of ConfigurationDesk 4.3

General

ConfigurationDesk 4.3 has a new software architecture with improved software handling and also some new features.

Assessing properties of elements

A central Properties Browser is now always available on the screen to display and configure properties, for example, of the signal chain elements. The Properties Browser replaces the properties dialogs of earlier software versions, where you had to open and close the dialogs for each element separately.
### Handling conflicts
The new concept of ConfigurationDesk allows very flexible configuration settings, with the effect that one setting might not match a setting at another place in the signal chain. These configuration conflicts are initially allowed, but before you build a real-time application, you have to resolve the conflicts to get proper build results. The new Conflicts Viewer shows all the conflicts and helps you to resolve them.

### Implementing tasks
Up to ConfigurationDesk 4.2, the implementation of tasks and components that influence the timing behavior of the executable application (= real-time application) is adopted from the behavior model (Simulink). ConfigurationDesk 4.3 now allows you to model executable applications and the tasks used within them very flexibly. For further information, refer to *Modeling Executable Applications and Tasks* ([ConfigurationDesk Real-Time Implementation Guide](#)).

### Undo/Redo
As of ConfigurationDesk 4.3, the software has undo/redo functionality. You can undo/redo actions or changes that you carried out beforehand.

### Rebuild detection for ControlDesk Next Generation
ConfigurationDesk provides specific build information so that ControlDesk Next Generation can detect whether new build results are available.

### Support of Simulink SLX model format
ConfigurationDesk 4.3 supports the Simulink SLX model format.

### Limitations concerning MATLAB compatibility
The following limitations apply when you work with MATLAB R2012a and R2012b:

**No incremental code generation for top-level models**
ConfigurationDesk does not support the Incremental Code Generation for Top-Level Models feature.

**Simulink Inport and Outport blocks used as I/O data interfaces**
If your model contains more than one periodic task, you must not use Simulink Inport and Outport blocks as I/O data interfaces. Otherwise, the build process is aborted with the following error message:

```
Compiling "E:\Prj1\App1\Build\myModel\Md1Code\myModel_dsrt\myModel.c"
myModel.c:
```
In function 'myModel_output0': myModel.c:185: error: 'tid'
undeclared (first use in this function)

**Only valid for MATLAB R2012a: No spaces in application path names** The application path name must not contain any spaces. Otherwise, the build process is aborted with an error message similar to the following one:  

```
Compiling "D:\dSPACE RCPHIL 7.4\MATLAB\DSRT\C\dsrt_mdflfcn.cpp"
dsrt_mdflfcn.cpp:19:21: error: rtmodel.h: No such file or directory
<...>
In file included from dsrt_mdflfcn.cpp:21:
dsrt_mdflfcn.h:34: error: variable or field
'APLlastApplStateStopped' declared void
<...>
ntox86-g++.exe: RCPHIL: No such file or directory
ntox86-g++.exe:
7.4\Work\Project_Test\Application\Build\dsmpblib_test\MdlCode":
No such file or directory
```

**Unsupported new features of MATLAB R2012b**

ConfigurationDesk does not support the following new features in MATLAB R2012b:

- Commenting out blocks
- Code generation for protected models
Migrating Projects from ConfigurationDesk 4.2 (on dSPACE Release 7.3) or Earlier

Reasons for migration

ConfigurationDesk 4.3 has a new software architecture with improved software handling and also some new features. To use projects and applications from earlier versions with ConfigurationDesk 4.3, you have to migrate them beforehand.

Running the migration

You can migrate a project created with ConfigurationDesk 4.2 or earlier automatically by opening it with ConfigurationDesk 4.3 as usual. ConfigurationDesk detects projects which need to be migrated and opens the migration dialog. When you click OK, migration starts and converts all the data and applications belonging to the project.

At the end of the migration process, the project opens automatically and you can work with it in ConfigurationDesk 4.3.

You cannot open and use a migrated project with ConfigurationDesk 4.2 or earlier.

During the migration process, a backup ZIP file is created for each application in the project. The ZIP files contain the old data and can be used for later restoration. Each `<Application name>.migrationbackup.zip` file is saved to the application root level.

Migration of Simulink models

Simulink models which are added to the ConfigurationDesk application are migrated the first time you open the model from ConfigurationDesk 4.3, for example, if you perform a model analysis.

Migration of custom function blocks

If you want to migrate a project which contains custom function blocks from demo models provided by dSPACE, an additional workaround is necessary. For details, refer to Migrating Custom Function Blocks Used in ConfigurationDesk 4.2 Projects or Earlier (ConfigurationDesk Real-Time Implementation Guide).
Changes after migration

Due to the changes in the software architecture, there are major and minor changes in the user interface and in handling. The minor changes do not really affect your work and its results. The major changes are described below.

Changes related to the Project Manager

The following changes affect projects and ConfigurationDesk applications:

- A ConfigurationDesk application no longer contains the Configuration application component. The CDS file which holds the data of the Configuration application component is no longer provided or supported. As of ConfigurationDesk 4.3, all application data (topology data and signal chain data) is stored in one internal zip file. You can import and export the data to reuse it in other projects.

- The topology files now are used only to exchange single topology data between different applications. The file name extensions of topology files have changed, for example, the file name extension of the hardware topology file has changed from HTF to HTFX.

Changes related to the model interface

The changes after migration are as follows:

- In the Model Port Block Library, the Trigger Event Port block has been replaced with the Runnable Function block, which exports a function-call subsystem as a runnable function. A runnable function groups together all the behavior model components that must be called for computing results in a specific task.

- In ConfigurationDesk:
  - You can access runnable functions (provided by the Runnable Function block in the behavior model) after model analysis. With the executable application and the task modeling feature, you can assign the runnable functions to a tasks (periodic or asynchronous tasks) for execution.

  **Runnable functions are not part of the model topology and therefore you cannot access them via the Model Browser.**

  - Trigger Event Port blocks that allow the transfer of asynchronous ConfigurationDesk function events to a Simulink behavior model do not exist any more.
- I/O function event ports (at function blocks) no longer exist. Access to the I/O events now is realized via the executable application and task modeling feature, where you can assign I/O events to tasks to trigger them.

**Limitations**

The following limitations exist for migrating projects:

**Migration of single topologies (HTF, DTF, MTF)** You cannot migrate single topologies, for example, a single device topology directly. If you want to migrate only one topology, you must add it in an empty application of ConfigurationDesk 4.2 or earlier. After this you can migrate the project containing the application.

**Obsolete and unresolved elements** Obsolete elements in a signal chain are not migrated. Unresolved elements of the hardware topology and the model topology are not migrated. However, unresolved elements of the device topology are migrated.

**Working view layouts** User-defined working view layouts for customized layout settings are not migrated and not supported with ConfigurationDesk 4.3.
## Container Manager

### New Features of the Container Manager 3.2

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Exporting V-ECU implementations in SystemDesk 3.2 and TargetLink** | In SystemDesk and TargetLink, you can generate V-ECUs for use with VEOS. You can export V-ECU implementations as containers. When you export a V-ECU implementation, a collection of files that belongs to the V-ECU implementation is gathered in a CTLGZ file archive. **Files in a V-ECU implementation** Three kinds of files can be part of a V-ECU implementation:  
- Configuration files (ARXML)  
- Code files (H, C)  
- A variable description file (A2L)  
For details, refer to *Export V-ECU Implementation (Container Management Document)*. |
| **Exchanging SWCs between SystemDesk 4.0 and TargetLink** | You can exchange software components between SystemDesk 4.0 and TargetLink using containers. You can use a container to transfer AUTOSAR files from SystemDesk 4.0 to TargetLink, either to start implementing a software component with TargetLink or to update an existing software component. After implementing or updating, you can export a container from TargetLink and import the relevant part of its contents in SystemDesk 4.0. For details, refer to *Exchanging Component Containers Between SystemDesk and TargetLink (Container Management Document)*. |
## ControlDesk 3.x

### Discontinuation of ControlDesk 3.x

ControlDesk 3.x will be 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](#) (ControlDesk Next Generation Migration Guide).

### Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of ControlDesk 3.7.4</td>
<td>58</td>
</tr>
<tr>
<td>Migrating to ControlDesk 3.7.4</td>
<td>58</td>
</tr>
</tbody>
</table>
New Features of ControlDesk 3.7.4

MicroAutoBox support

The Platform Manager now displays FPGA piggyback and IP modules connected to MicroAutoBox.

Migrating to ControlDesk 3.7.4

Reusing CAN/LIN bus configurations last saved in ControlDesk 3.7.2 or earlier

Suppose you worked in ControlDesk 3.x with a bus configuration (XML file) based on a real-time application built with RTI CAN MultiMessage Blockset or RTI LIN MultiMessage Blockset from dSPACE Release 7.2 or earlier. When you reuse this bus configuration in ControlDesk 3.7.4 after rebuilding the application with the RTI MultiMessage Blockset from dSPACE Release 7.3, some functions such as replay and variant handling do not work (due to a changed build process).

To reuse your old bus configuration, you must migrate it to ControlDesk 3.7.4. ControlDesk provides the DSMigrateBusCfg.exe command line tool for this. You can find the tool at %dSPACE_ROOT%\ControlDesk\bin after installation of ControlDesk.

To migrate a bus configuration, perform the following steps:

1. Open a Command Prompt window. For example, on the Start menu in Windows® XP, select Run, enter cmd and click OK.
2. Enter DSMigrateBusCfg InputFile [OutputFile].
   - InputFile is the old bus configuration file, whose file name extension must be XML. This parameter is mandatory.
   - OutputFile is the new bus configuration file and is an optional parameter. If you specify the OutputFile parameter, it must have the XML file name extension. If you do not specify the OutputFile parameter, the new bus configuration has the name of the old bus configuration with the suffix _new.
ControlDesk Next Generation

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>New Features of ControlDesk Next Generation (ControlDesk 4.3)</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrating to ControlDesk Next Generation (ControlDesk 4.3)</td>
<td>75</td>
</tr>
</tbody>
</table>

Information in other sections

ControlDesk Next Generation Migration Guide
Explains migration from CalDesk/ControlDesk 3.x to ControlDesk Next Generation (ControlDesk 4.3).
New Features of ControlDesk Next Generation (ControlDesk 4.3)

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>New General Features (ControlDesk 4.3)</td>
<td>60</td>
</tr>
<tr>
<td>New Features of Platform Management and Platforms/Devices (ControlDesk 4.3)</td>
<td>62</td>
</tr>
<tr>
<td>New Variable Management Features (ControlDesk 4.3)</td>
<td>64</td>
</tr>
<tr>
<td>New Instrument Features (ControlDesk 4.3)</td>
<td>65</td>
</tr>
<tr>
<td>New Measurement and Recording Features (ControlDesk 4.3)</td>
<td>70</td>
</tr>
<tr>
<td>New Bus Navigator Features (ControlDesk 4.3)</td>
<td>70</td>
</tr>
<tr>
<td>New Data Set Management Features (ControlDesk 4.3)</td>
<td>70</td>
</tr>
<tr>
<td>New ECU Diagnostics Features (ControlDesk 4.3)</td>
<td>72</td>
</tr>
<tr>
<td>New Automation Features (ControlDesk 4.3)</td>
<td>73</td>
</tr>
</tbody>
</table>

New General Features (ControlDesk 4.3)

VEOS support

ControlDesk supports VEOS, the successor to the dSPACE Offline Simulator. The new VEOS support feature is:


dSPACE Release 7.4 provides VEOS as a new software product to perform offline simulation on your host PC. However, you must install an add-on to dSPACE Release 7.4 to work with VEOS in connection with ControlDesk Next Generation (ControlDesk 4.3). The add-on is available at http://www.dspace.com/go/VEOS-AddOn.
Camera interface

ControlDesk 4.3 provides a new interface to access video cameras connected to the ControlDesk PC. The interface lets you display, record, and replay video data synchronously with data from vehicle buses, ECUs, dSPACE RCP systems, etc.

ControlDesk's camera interface supports all cameras that use DirectShow. The supported cameras are usually Plug and Play cameras connected via USB or Ethernet.

You can save recorded video data as an IDF or AVI file.

The camera interface consists of:
- A Video Capturing device. Refer to Configuring a Video Capturing Device (ControlDesk Next Generation Basic Practices Guide).

Supported CAN interfaces

ControlDesk now also supports the following CAN interfaces from Vector Informatik:
- VN16xx
- CANcardXLe

Refer to Supported CAN Interfaces (ControlDesk Next Generation Basic Practices Guide).
New Features of Platform Management and Platforms/Devices (ControlDesk 4.3)

Information in this topic

New platforms/devices on page 62
VEOS platform on page 62
Video Capturing device on page 62
Automatic reload of variable description after rebuild on page 62
Detecting terminated real-time applications on page 63
Clearing the system on page 63
SCALEXIO Processing Unit platform: Switching between assembly view and network view on page 63
Multiprocessor System platform: Changed configuration process on page 63
MicroAutoBox platform: Display of connected piggyback modules on page 63
DS230x: Registration via processor board on page 64

New platforms/devices

ControlDesk Next Generation now also supports the following platforms/devices:

**VEOS platform**  ControlDesk’s new VEOS platform lets you configure and control an offline simulation application running in VEOS and access the application’s environment VPU. The VEOS platform replaces the Offline Simulator platform from ControlDesk 4.2.1.

**Video Capturing device**  ControlDesk’s new Video Capturing device lets you capture video signals synchronously to signals from other platforms/devices.

Refer to Configuring a Video Capturing Device (ControlDesk Next Generation Basic Practices Guide).

Automatic reload of variable description after rebuild

ControlDesk can now detect the rebuild of a real-time application if it was built using an implementation software (RTI or ConfigurationDesk) version dSPACE Release 7.4 or later. You can specify to reload the variable description file automatically in such cases.

Refer to Variables Page (ControlDesk Next Generation Reference).
Detecting terminated real-time applications

When you start online calibration, ControlDesk automatically detects whether a real-time application has been terminated. You cannot run a measurement for a terminated application. You can specify whether to reload the application.

Refer to Basics on Real-Time Applications (ControlDesk Next Generation Basic Practices Guide).

Clearing the system

ControlDesk lets you clear the system by erasing the recent hardware configuration. Then it refreshes the interface connections between the host PC and the hardware to reinitialize the Platform/Device Manager.

When the system is cleared, ControlDesk creates a copy of the recent hardware configuration. To recover the system, you can import the copy again via the Manage Recent Hardware Configuration dialog.

Refer to Clear System (ControlDesk Next Generation Reference).

SCALEXIO Processing Unit platform: Switching between assembly view and network view

You can now switch between an assembly view and a network view for the SCALEXIO Processing Unit platform:

- Assembly view: To display a component-based view of the registered hardware in the Platform/Device Manager. Refer to Assembly View (ControlDesk Next Generation Reference).

- Network view: To display a network-based view of the registered hardware in the Platform/Device Manager. Refer to Network View (ControlDesk Next Generation Reference).

Multiprocessor System platform: Changed configuration process

The way you configure a Multiprocessor System platform has changed in ControlDesk 4.3.

For instructions, refer to How to Register a Platform (ControlDesk Next Generation Basic Practices Guide).

For information on the Multiprocessor System platform and on assigning registered hardware to a Multiprocessor System platform, refer to Basics on Multiprocessor System Platforms (ControlDesk Next Generation Basic Practices Guide).

MicroAutoBox platform: Display of connected piggyback modules

The Platform/Device Manager now displays piggyback modules connected to MicroAutoBox.
DS230x: Registration via processor board
As of ControlDesk 4.3, a DS230x board can be registered only by registering the processor board to which the DS230x board is connected. When you register the processor board, a connected DS230x board is detected automatically.
Refer to Register Platform (ControlDesk Next Generation Reference).

New Variable Management Features (ControlDesk 4.3)

Automatic reload of variable description after rebuild
ControlDesk can now detect the rebuild of a real-time application if it was built using an implementation software (RTI or ConfigurationDesk) version dSPACE Release 7.4 or later. You can specify to reload the variable description file automatically in such cases.
Refer to Variables Page (ControlDesk Next Generation Reference).

Combined variable filter
To filter the variable list in ControlDesk's Variable Browser, you can now define a filter that combines several filter conditions.
You can define a combination of the following filter conditions:
- Wildcard filter
- Variable state
- Variable type
You can also specify filter groups to define cascaded filters.

For instructions, refer to How to Filter the Variable List (ControlDesk Next Generation Basic Practices Guide)

New Instrument Features (ControlDesk 4.3)

Information in this topic

- Avionics instruments on page 66
- Airspeed Indicator on page 66
- Altimeter on page 66
- Artificial Horizon on page 67
- Heading Indicator on page 67
- New Video Monitoring instrument on page 68
- Importing layouts as references on page 68
- Assigning instrument properties on page 68
- Locking layout editing for each layout individually on page 69
- Discontinuation of the Configure Variables dialog on page 69
- Further instrumentation features on page 69
ControlDesk now provides the following Avionics instruments to visualize variables of a simulated aircraft.

<table>
<thead>
<tr>
<th>Avionics instruments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Airspeed Indicator</strong></td>
<td>To indicate the airspeed of a simulated aircraft.</td>
</tr>
<tr>
<td><strong>Altimeter</strong></td>
<td>To indicate the altitude of a simulated aircraft.</td>
</tr>
</tbody>
</table>

Refer to [Airspeed Indicator](#) or [Altimeter](#) in the [ControlDesk Next Generation Reference](#).

![Airspeed Indicator Diagram](image)

- Visible neighbor
- Airspeed
- Airspeed indicator scale with colored bands
- Magnifier

Refer to [Altimeter](#) in the [ControlDesk Next Generation Reference](#).

![Altimeter Diagram](image)

- Visible neighbor
- Magnifier
- Altimeter scale

**Caption with BLOCK and VARIABLE macro**
**Artificial Horizon**

To indicate a simulated aircraft's pitch and roll.

Refer to *Artificial Horizon* (ControlDesk Next Generation Reference).

**Heading Indicator**

To indicate the direction a simulated aircraft is headed in.

Refer to *Heading Indicator* (ControlDesk Next Generation Reference).
ControlDesk Next Generation now provides the Video Monitoring instrument for displaying video data.

The illustration below shows the Video Monitoring instrument displaying recorded video data:

Refer to Video Monitoring Handling ([link] ControlDesk Next Generation Basic Practices Guide).

You can import a layout as a local working copy with a reference to an external layout file (LAX file) by importing the external layout as a link. You can then synchronize the layout settings via the context menu of the new experiment layout.

Refer to:
- Import Layout(s) ([link] ControlDesk Next Generation Reference)
- Synchronize (Layout) ([link] ControlDesk Next Generation Reference)

You can now easily assign properties of one instrument to another (if the properties are common to both instruments).

Refer to Assign Properties ([link] ControlDesk Next Generation Reference).
<table>
<thead>
<tr>
<th><strong>Locking layout editing for each layout individually</strong></th>
<th>You can now lock layout editing for each layout individually. Up to now, this was possible for the entire experiment only. Refer to Locked Mode ([ControlDesk Next Generation Reference]).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Discontinuation of the Configure Variables dialog</strong></td>
<td>The Configure Variables dialog has been discontinued in ControlDesk 4.3. The dialog was used to select instruments and layouts for visualizing variables. To visualize variables in ControlDesk 4.3, select them in the Variable Browser and press Enter. The variables are placed on the currently active layout and connected to their default instrument.</td>
</tr>
</tbody>
</table>
| **Further instrumentation features**                   | Further instrumentation features:  
  - Improvements for instruments that use a scale (Knob, Gauge, ...)  
  - Table Editor: Support for the extended and fixed scaling modes for data axes  
    - Extended mode: The data axis can only grow.  
    - Fixed mode: The range of the data axis is set to the values of the Min and Max properties.  
  Refer to Axes Properties (Table Editor) ([ControlDesk Next Generation Reference]).  
  - Multiselecting layouts for export  
  Refer to Export ([ControlDesk Next Generation Reference]).  
  - Configuring the instrument selection border style  
    ![Selector border](image)
    Refer to Instruments Page ([ControlDesk Next Generation Reference]). |
New Measurement and Recording Features (ControlDesk 4.3)

Support of ASAM MDF 4.0  ControlDesk now supports the ASAM MDF 4.0 format (MF4 file name extension) for importing and exporting measurement data.

New Bus Navigator Features (ControlDesk 4.3)

ControlDesk’s Bus Navigator now provides the following new features:

Features for CAN
- Generation of Bus Instruments (TX Type) in connection with PC-based CAN interfaces such as DCI-CAN1

Features for FlexRay
- Generation of Bus Instruments (RX Type) in connection with PC-based FlexRay interfaces
- Filtered monitoring and logging in connection with PC-based FlexRay interfaces

Features for LIN
- Generation of Bus Instruments (RX Type) in connection with PC-based LIN interfaces
- Filtered/unfiltered monitoring and logging in connection with PC-based LIN interfaces

Demo
ControlDesk now provides a Bus Navigator demo. Refer to Bus Navigator Demo (ControlDesk Next Generation Basic Practices Guide).

New Data Set Management Features (ControlDesk 4.3)

Restoration of data sets after reload
If you reload the variable description (SDF file) of a dSPACE platform, ControlDesk maintains the data set assignment and tries to reestablish all parameter connections.

Refer to Reload Variable Description (ControlDesk Next Generation Reference).
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifying data sets to be downloaded on online calibration start</td>
<td>ControlDesk now lets you select one or more data sets / sub data sets to be downloaded to a platform on online calibration start. If you select more than one data set / sub data set to download on online calibration start, the data sets / sub data sets are downloaded one after the other according to their order in the Project Manager. Refer to How to Select Additional Data Sets for Downloading on Online Calibration Start (ControlDesk Next Generation Basic Practices Guide).</td>
</tr>
<tr>
<td>Creating data sets for devices that have no memory segment definition</td>
<td>ControlDesk now lets you create data sets for devices that have no memory segment definition. Refer to Create Data Set (ControlDesk Next Generation Reference).</td>
</tr>
<tr>
<td>Creating an application image</td>
<td>If a platform supports creating a new application image, you can now upload the real-time image to the hardware in one step. Refer to How to Create an Application Image (ControlDesk Next Generation Basic Practices Guide).</td>
</tr>
<tr>
<td>Automatic EPK check on import data set from HEX file</td>
<td>When you import a HEX file (HEX, S19, MOT), ControlDesk checks the consistency of the A2L file and the imported data set (EPK check), if possible. Refer to Consistency Checks (EPK Checks) (ControlDesk Next Generation Basic Practices Guide).</td>
</tr>
<tr>
<td>EPK check for MMF-based data sets</td>
<td>For HEX files (HEX, S19, MOT, ControlDesk internal file format: MMF), you can initiate a consistency check (EPK check) manually. Refer to Check EPK (ControlDesk Next Generation Reference).</td>
</tr>
<tr>
<td>Filtering data set parameters by memory segments</td>
<td>In ControlDesk’s Data Set Manager, you can now reduce the numbers of displayed parameters by choosing the Memory Segments Filter prefilter. Refer to How to Filter a Data Set (ControlDesk Next Generation Basic Practices Guide).</td>
</tr>
</tbody>
</table>
New ECU Diagnostics Features (ControlDesk 4.3)

Working with several ODX databases (only for ECU Diagnostics v2.0.2 devices)
If you work with an ECU Diagnostics v2.0.2 device, you can specify several ODX diagnostics databases for it and switch between them. Working with several ODX databases enables you to check a further ODX diagnostics database within an existing experiment without having to modify or copy the experiment. After a successful test, you can easily make the new diagnostics database the only diagnostics database for the device by removing the other databases.

Refer to Basics of the ECU Diagnostics Device (ControlDesk Next Generation Basic Practices Guide).

Optimizing the database
If you work with an ECU Diagnostics v2.0.2 device, you can use a proprietary binary format instead of ODX data for the diagnostics database. Using binary files speeds up experiment loading and lets you work with large ODX databases, since memory usage is reduced. However, transforming the ODX data into the binary format takes some time, so you should think carefully whether to use ODX data or the binary format with your experiment.

Using the binary format is useful for experiments
- That are frequently loaded, and whose diagnostics database is modified only rarely.
- That require large ODX databases.

For details, refer to How to Configure an ECU Diagnostics Device (ControlDesk Next Generation Basic Practices Guide).

Enhancements to ECU flash memory programming via ECU diagnostics
The dialog for programming the flash memory of an ECU via a diagnostic protocol is enhanced.

Refer to How to Program the ECU Flash Memory via a Diagnostic Protocol (ControlDesk Next Generation Basic Practices Guide).

Measurement and calibration of diagnostic variables: Support for TEXTTABLES
For TEXTTABLES in an ODX database, ControlDesk now generates variables with a COMPU_VTAB conversion table when you generate a variable description from the ODX database. When you visualize such a variable, for example, in a Variable Array, you can select the converted value in the instrument.
Enhanced demo
The ECU Diagnostics demo is enhanced.
Refer to *ECU Diagnostics Demo* ([ControlDesk Next Generation Basic Practices Guide](#)).

### New Automation Features (ControlDesk 4.3)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Automating direct variable access** | ControlDesk’s automation interface now lets you read and write variables directly without having to use instruments. For example, you can read and write the values of scalar variables (e.g., scalar parameters and measurement variables) and multidimensional variables (e.g., value blocks, maps, curves and measurement arrays).

Refer to *Automating Direct Variable Access* ([ControlDesk Next Generation Advanced Practices Guide](#)). |
| **Automating the Bus Navigator** | ControlDesk’s automation interface now lets you automate bus communication monitoring, logging, and replaying. Refer to *Automating the Bus Navigator* ([ControlDesk Next Generation Advanced Practices Guide](#)).

Your ControlDesk installation also provides a demo for automating the Bus Navigator in the `.\Demos\ToolAutomation\<ProgrammingLanguage>` folder. |
| **Enhancements to automating ECU diagnostics** | Reading DTCs from the ECU fault memory and environment data for a specific DTC via automation has been simplified.

Refer to *Automating ECU Diagnostics Tasks* ([ControlDesk Next Generation Advanced Practices Guide](#)). |
| **Automating the ControlDesk NG Properties dialog** | You can now specify most of the settings of the ControlDesk NG Properties dialog via automation. Refer to the *ControlDesk Next Generation API Reference*. |
| **Assigning external scripts to events** | You can now assign external scripts to ControlDesk events. For details, refer to *Importing Python Scripts from the Active Project or Experiment to the Event Handler* ([ControlDesk Next Generation Advanced Practices Guide](#)). |
Automating groups in an experiment

You can now add groups to an experiment via automation. Groups are structuring elements that are only displayed in the project tree but not stored in the file system.

For details, refer to Add Group (ControlDesk Next Generation Reference).
Migrating to ControlDesk Next Generation (ControlDesk 4.3)

Discontinuation of the ECU Diagnostics v2.0.1 Device
The ECU Diagnostics v2.0.1 device (supporting ASAM MCD-3 D V2.0.1) will be discontinued in 2013. This means you must migrate to the ECU Diagnostics v2.0.2 device (supporting ASAM MCD-3 D V2.0.2), the successor to the ECU Diagnostics v2.0.1 device.

Migrating from ControlDesk 4.2.1
To migrate from ControlDesk 4.2.1 to ControlDesk 4.3 and reuse existing experiments, you do not have to carry out any migration steps.

Migrating from CalDesk, ControlDesk 3.x, or ControlDesk Next Generation 4.0/4.1
To migrate from CalDesk, ControlDesk 3.x, or ControlDesk Next Generation 4.0/4.1 and reuse existing experiments, you may 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 1.3

<table>
<thead>
<tr>
<th>Enhanced platform support</th>
<th>VEOS as the new offline simulation platform is supported by dSPACE HIL API .NET using the MAPort implementation.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dSPACE Release 7.4 provides VEOS as a new software product to perform offline simulation on your host PC, but does not include VEOS support by HIL API .NET 1.3. However, you must install an add-on to dSPACE Release 7.4 to work with VEOS in connection with HIL API .NET 1.3. The add-on is available at <a href="http://www.dspace.com/go/VEOS-AddOn">http://www.dspace.com/go/VEOS-AddOn</a>.</td>
</tr>
</tbody>
</table>

For further information on the dSPACE HIL API .NET implementation, refer to *dSPACE HIL API .NET Implementation - Notes*. 
dSPACE FlexRay Configuration Package

New Features of dSPACE FlexRay Configuration Package 3.0

<table>
<thead>
<tr>
<th>FlexRay Configuration Package</th>
<th>Support of PDU transmission modes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dSPACE FlexRay Configuration Package 3.0 now supports transmission modes (as defined in AUTOSAR 3.1.4 and later). It has two transmission modes called True and False, which are assigned to the PDUs. A specific timing is defined in the communication cluster file and assigned to each transmission mode. The FlexRay Configuration Package also provides the User-Defined transmission mode for creating an additional timing based on the LPDU timing. You can switch between the PDU transmission modes during run time. You can work with different PDU transmission modes in connection with configurations based on the following communication cluster files:</td>
</tr>
</tbody>
</table>

- FIBEX 3.0 files
- FIBEX 3.1 files |
- FIBEX 3.1.1 files
- AUTOSAR system description files based on AUTOSAR 3.1.4, 3.2.1, or 3.2.2

For further information, refer to How to Configure PDU Transmission Modes (FlexRay Configuration Tool Guide).

**Update bit behavior according to AUTOSAR**

The update bit behavior has changed in dSPACE FlexRay Configuration Package 3.0. The update bit behavior is now defined on the basis of AUTOSAR specifications. Up to dSPACE FlexRay Configuration Package 2.7, the update bit of a TX PDU was set to 1 each time that the payload of the PDU had changed. As of dSPACE FlexRay Configuration Package 3.0, the update bit is set to 1 each time that the IPDU data is packed, even if no new data is available. IPDU data is packed according to IPDU timing, that is, the update bit value is set to 1 each time that the IPDU is sent.

Update bit manipulation has also changed: It can now be performed for a cyclic IPDU only when the IPDU is sent, not each time that the LPDU is sent, because this would be an indirect manipulation of the PDU timing.

**Support of static event PDUs**

With dSPACE FlexRay Configuration Package 3.0, static event PDUs are no longer treated like cyclic PDUs. A trigger for sending static event PDUs is available. The trigger is part of the PDU bus of the RTIFLEXRAYCONFIG PDU TX or FLEXRAYCONFIG PDU TX block.

**Enhancements to static TX PDUs**

With older versions of the dSPACE FlexRay Configuration Package, static TX PDUs were packed according to the LPDU timing, and any cycle time definition was not evaluated. dSPACE FlexRay Configuration Package 3.0 takes into account the cycle time of static TX PDUs, which are now packed according to their cycle times. Thus, the update bit of a static TX PDU is set to 1 only in the cycles in which the PDU is newly packed.

**Support of AUTOSAR System Template 3.2.2**

The FlexRay Configuration Tool now also supports the AUTOSAR System Template based on AUTOSAR Release 3.2.2 for describing FlexRay networks.
**Alias name for physical bus access** The FlexRay Configuration Tool lets you specify an alias for the physical FlexRay bus. The specified alias name is used in ControlDesk Next Generation’s Bus Navigator. It is displayed in the Bus Navigator tree next to the respective controller icon.

For further information, refer to General Page (FlexRay Configuration Tool Reference).

**Assigning tasks to timetables** The FLEXRAYCONFIG TIMETABLE TASK blocks that were used to assign tasks to timetables in previous versions of the FlexRay Configuration Blockset are obsolete. The block is still visible in the FlexRay Configuration Blockset for migration purposes, but it no longer provides functionality.

FlexRay Configuration Blockset 2.0 now uses Runnable Function blocks of the Model Port Block Library to assign tasks to timetables. Runnable Function blocks are added to the automatically generated FlexRay model during the generation process. A Runnable Function block is named after its task. The block lets you view the port configuration settings and displays the block parameters specified during the generation process. When you update the FlexRay blocks in your Simulink model, the Runnable Function blocks are included in the automatic update process.

For further information, refer to the Simulink Model Block Reference.
# Migrating to dSPACE FlexRay Configuration Package 3.0

<table>
<thead>
<tr>
<th>Migrating models based on dual-channel configurations with different timings for the channels</th>
<th>After a FlexRay model based on a dual-channel configuration is migrated, the default transmission mode is assigned to all PDUs by the FlexRay Configuration Tool regardless of channel. If your model is based on a dual-channel configuration with different timings for the two channels, you must set the transmission mode for the PDUs separately for each channel during run time. Switch the transmission mode where necessary.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed update bit behavior</td>
<td>The update bit behavior has changed in dSPACE FlexRay Configuration Package 3.0, which causes differences in the run-time behavior when you reuse existing FlexRay models. To make the new update bit behavior available to an existing FlexRay model, you must generate new configuration data and code for the FlexRay configuration with the FlexRay Configuration Tool and then update your Simulink model and start the build process.</td>
</tr>
</tbody>
</table>
| Migrating models with static event PDUs | The handling of static event PDUs has changed. Now a trigger is used to send static event PDUs. The trigger is part of the PDU bus of the RTIFLEXRAYCONFIG PDU TX or FLEXRAYCONFIG PDU TX block.  
- If you use PDU-based modeling, you must add the TxTrigger inport of the TX PDU block manually. There is no automatic model update. You can copy the trigger part from the newly generated PDU mapping subsystem.  
- With signal-based modeling, static event PDUs are not supported. Static event PDUs cannot be used, that is, they are not sent and their update bits remain 0. |
| Changed transmission times for static TX PDUs | The transmission times of static TX PDUs have changed. Static TX PDUs are now sent according to their cycle times, if available. The changed transmission times can cause differences in the run-time behavior when you reuse existing FlexRay models containing static TX PDUs with cycle times. To apply the new transmission times to your model, you must first generate new code with the FlexRay Configuration Tool, and then update your Simulink model and start the build process. |
dSPACE Python Extensions

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of dSPACE Python Extensions 1.3</td>
</tr>
<tr>
<td>Migrating to dSPACE Python Extensions 1.3</td>
</tr>
</tbody>
</table>

New Features of dSPACE Python Extensions 1.3

dSPACE Platform Management API

With the dSPACE Platform Management API, you can access your platform via script. For example, you can register a platform and load a real-time application. It is not necessary to use a program such as AutomationDesk or ControlDesk Next Generation for platform management.

For further information, refer to dSPACE Platform Management API Reference.
VEOS as the new offline simulation platform is supported by dSPACE Python Extensions using the HIL API Python implementation for the MAPort.

For further information on the dSPACE HIL API Python Implementation, refer to dSPACE HIL API Python Implementation - Notes.

Migrating to dSPACE Python Extensions 1.3

Registering the platform

To use a platform with rplib2 or the dSPACE HIL API Python Implementation of the MAPort, it has to be registered via the dSPACE Platform Management API or ControlDesk Next Generation or AutomationDesk. Platforms registered via ControlDesk 3.x are ignored.
ECU Interface Manager

Where to go from here

Information in this section

New Features of ECU Interface Manager 1.2  85
Migrating to ECU Interface Manager 1.2  87

New Features of ECU Interface Manager 1.2

Automating the ECU Interface Manager

You can automate the application configuration and code generation via the new ECU Interface Manager automation interface.

The main functions of the automation interface are:

- Managing, i.e., opening, saving, and closing a project
- Importing software modules (ECU applications and dSPACE bypassing services)
- Adding bypassing service calls to the application
- Exporting the application

You can automate the ECU Interface Manager in two ways:

- Via an interface that allows you to automate the ECU Interface Manager via XML files or XML fragments. The interface can be used via command line.
- Via an M API that allows you to automate the ECU Interface Manager from MATLAB.

For details, refer to Automating the ECU Interface Manager (ECU Interface Manager Guide).
### Password protection for ECU applications

The encryption tool of the ECU Interface Manager now has a password protection option so that suppliers can give software modules better protection when shipping them to their customers. Software module description files can be encrypted and password-protected.

For details, refer to [Encrypting a Software Module Description File](#) (*ECU Interface Manager Guide*).

### Disabling function calls and write accesses

To improve performance, you can disable function calls and write accesses to variables. The disabled parts of the code are referenced in the exported A2L file.

- You can disable functions in the ECU code to improve the ECU’s runtime behavior. You can use the saved processing time for the integration of new internal bypass functions.
- You can disable write accesses to ECU variables. This allows you to specify the values of these variables by using a measurement and calibration tool such as ControlDesk Next Generation or by using a bypass function.

For details, refer to [How to Disable Code Items in the ECU Application](#) (*ECU Interface Manager Guide*).

### Backup current project

You can now back up your current project from the ECU Interface Manager. The project files are archived with relative paths, so they can be extracted anywhere. If a ZIP archive contains a file from a different file system, the absolute path is stored.

For details, refer to [Backup Project](#) (*ECU Interface Manager Reference*) and [Open Project from Backup](#) (*ECU Interface Manager Reference*).

### LOG file

You can open the ECU Interface Manager’s LOG file directly from the user interface of the ECU Interface Manager.

For details, refer to [Log File](#) (*ECU Interface Manager Reference*).
Migrating to ECU Interface Manager 1.2

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

In ECU Interface Manager 1.2, 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 1.2, 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 1.2 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 with which you can work in the former version of ECU Interface Manager.
Model Compare

New Features of Model Compare 2.3

Enhanced tree view display options

Model Compare now offers even more options for showing/hiding blocks, lines, and annotations in the tree view of the Model Navigator. The improvement also includes shortcut access to the most important options. For details, refer to the description of the Block Display (Model Compare Reference), Line Display (Model Compare Reference), and Annotation Display (Model Compare Reference) toolbar buttons.

Tree view display considering comments

Model Compare now can show/hide nodes in the tree view depending on whether the nodes are commented (including shortcut access), refer to the Node Display Concerning Comments (Model Compare Reference) toolbar button.
Highlighting with smarter window opening

During recursive highlighting, Model Compare no longer opens each of the subsystems where blocks or lines are highlighted, but only the nearest common subsystem is displayed in Simulink. Refer to How to Visualize Model Differences in MATLAB (Model Compare Guide).

Savable highlighting

As an alternative to highlighting, you can now colorize block differences for one or more selected subsystems or blocks as a kind of permanent or savable highlighting. If subsystems are selected, all the child subsystems can also be colorized recursively. Colorization can be saved with the Simulink model but it cannot be removed (undone) from within Model Compare. Refer to Colorize Differences (Model Compare Reference) and Colorize Differences Recursively (Model Compare Reference).

Support of SLX file format

Model Compare now supports the SLX Simulink model file format. You can dump, compare, and merge SLX files in the same way as with MDL model files.

Compatibility with TargetLink

Model Compare 2.3 is fully compatible with TargetLink 3.4.

Migration to Model Compare 2.3

User preferences and comparison settings

User preferences and comparison settings are stored separately from the tool. Thus, Model Compare 2.3 does not make use of the user preferences and comparison settings previously specified with Model Compare 2.0/2.1/2.2, but uses the defaults. However, user preferences and comparison settings that have been exported with Model Compare 2.0/2.1/2.2 can be imported by Model Compare 2.3.

Model Compare 2.3 lets you specify more user preferences and comparison settings than is possible with Model Compare 2.0 or older. For details refer to User Preferences Dialog (Model Compare Reference).
ModelDesk

New Features of ModelDesk 2.6

New supported platform

dSPACE Release 7.4 provides VEOS as a new software product to perform offline simulation on your host PC.

However, you must install an add-on to dSPACE Release 7.4 to work with VEOS in connection with ModelDesk 2.6. The add-on is available at http://www.dspace.com/go/VEOS-AddOn.

Support of Simulink SLX model format

ModelDesk now supports the new Simulink SLX model format. With MATLAB R2012b, models containing ASM blocks can be opened or saved in MDL or SLX format. Both formats can be used in ModelDesk projects and for custom libraries.

Extended tool automation

The tool automation interface of ModelDesk is extended.

- You can now save, delete, link or download parameter files via the automation interface.
- You can reconfigure parameter sets via the automation interface, for example, changing the transmission from manual to automatic.
MotionDesk

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of MotionDesk 3.1</td>
<td>94</td>
</tr>
<tr>
<td>Migrating to MotionDesk 3.1</td>
<td>95</td>
</tr>
</tbody>
</table>
New Features of MotionDesk 3.1

Scene Editor
The Scene Editor has a uniform scale mode.

Visual enhancements
The visualization is enhanced.
- MotionDesk has new settings for atmospherics: rain and snow fall.

- A new fullscreen mode shows the scene without any frames.
- Friction maps modeled with ModelDesk’s Road Generator can be visualized.

New supported platforms or connection
The following features are new for the data source:
- MicroAutoBox II is supported as a simulation platform.
- Platforms with a slot CPU that are connected to the host PC via Ethernet are supported.

The frame rate may be slow for both features.

Automation interface
MotionDesk has an automation interface which lets you use scripts to control MotionDesk. It provides the following features:
- Projects and experiments
  - Opening and saving project and experiment
Library
- Importing 3-D objects into the customer object library

Scene
- Adding and removing movable objects
- Reading and setting properties of movable objects
- Adding and removing static objects
- Reading and setting properties of static objects
- Reloading the scene

Visualization
- Starting and stopping the animation
- Setting the window mode (normal or full screen)

**Related topics**

<table>
<thead>
<tr>
<th>Basics</th>
<th>HowTos</th>
</tr>
</thead>
<tbody>
<tr>
<td>✷ Automating MotionDesk (MotionDesk Guide)</td>
<td>✷ How to Visualize Rain and/or Snow Fall (MotionDesk Guide)</td>
</tr>
</tbody>
</table>

**Migrating to MotionDesk 3.1**

**Migrating from MotionDesk 2.1.6 and earlier**

From 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**

As MotionDesk uses another format for the 3-D objects, the scenes and custom 3-D objects must be migrated. For details, refer to **Migrating from MotionDesk 2.2.1 and Lower (MotionDesk Guide)**.
RapidPro Control Unit RTI Blockset

New Features of the RapidPro Control Unit RTI Blockset

| Arbitrary crankshaft wheel design | You can specify a crankshaft wheel via a wavetable comprising 7200 data points (corresponding to an angular resolution of 0.05°).
|                                | Specifying a crank wheel via wavetable is advantageous because:
|                                | - The gaps and teeth of a crankshaft wheel can have arbitrary lengths.
|                                | - Asymmetric crankshaft wheels can be used.
|                                | - There is no need to specify a camshaft evaluation segment.
|                                | - Synchronization can be achieved faster.
|                                | For details on this new feature, refer to Basics on Crankshaft Wheel Specification (RapidPro System – I/O Subsystem MPC565 Implementation Features).

| Support of reverse crankshaft rotation | A reverse crankshaft rotation does not necessarily lead to the loss of synchronization. This is possible, if the crankshaft wheel is specified via wavetable and you have enabled the reverse crankshaft operation mode. Sophisticated information on the engine status is returned.
|                                          | For details on this new feature, refer to Basics on the Engine Status (Crank Wavetable) (RapidPro System – I/O Subsystem MPC565 Implementation Features).
Smoothed speed measurement

If the crankshaft wheel is specified via wavetable, you can smooth speed measurement by applying a recursion. The measured speed value and the last speed result are used to calculate the engine speed. The influence of the last speed result, i.e., the degree of smoothness, is defined by a weighting factor. In addition, speed measurement is more precise if the crankshaft wheel is specified via wavetable and the speed changes abruptly, for example, at a gap.

For details on this new feature, refer to *Basics on Speed Measurement (Crank Wavetable)* (RapidPro System – I/O Subsystem MPC565 Implementation Features).
Real-Time Testing

Where to go from here

Information in this section

- New Features of Real-Time Testing 2.0
- Migrating to Real-Time Testing 2.0

New Features of Real-Time Testing 2.0

New supported platform

dSPACE Release 7.4 provides VEOS as a new software product to perform offline simulation on your host PC.

However, you must install an add-on to dSPACE Release 7.4 to work with VEOS in connection with Real-Time Testing (RTT) 2.0. The add-on is available at http://www.dspace.com/go/VEOS-AddOn.

Migrating to Real-Time Testing 2.0

Limitation of Real-Time Testing caused by the installation

Real-Time Testing 1.6 is part of Release 6.5. If you activate Release 6.5, all tools which use Real-Time Testing from the Python interpreter use Real-Time Testing 1.6 regardless of the settings in the dSPACE Installation Manager. This also applies to dSPACE tools installed with Release 7.4.

Since Release 6.6 (Real-Time Testing 1.7), this limitation does not exist.
# RTI/RTI-MP and RTLib

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

<table>
<thead>
<tr>
<th>Supported new features introduced with MATLAB R2012a</th>
<th>RTI and RTI-MP support the following new features introduced with MATLAB R2012a:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Incremental code generation for top-level models</td>
</tr>
<tr>
<td></td>
<td>Simulink SLX model format</td>
</tr>
</tbody>
</table>

| Possible problems with incremental code generation | With dSPACE Release 7.4, RTI supports the Incremental Code Generation for Top-Level Models feature introduced with MATLAB R2012a. If no changes in the model are detected, the build process skips code generation for the model and recompiles the code which was generated in the last build process. This can cause problems if the RCP & HIL environment has changed since the last code generation. For example, the RCP & HIL environment is changed if you switch to another RCP & HIL installation. For further information, refer to Troubleshooting (RTI and RTI-MP Implementation Guide). |
Unsupported new features of MATLAB R2012b

RTI and RTI-MP do not support the following new features in MATLAB R2012b:

- Commenting out blocks
- Code generation for protected models

rti_mdlversionget command discontinued

The `rti_mdlversionget` command is obsolete and has been removed. It is not compatible to the new SLX model format.

Rebuild detection for ControlDesk Next Generation

RTI and RTI-MP provide specific build information so that ControlDesk Next Generation can detect whether a new application file is available.

MicroAutoBox

MicroAutoBox has some hardware and software enhancements.

**Acceleration sensor**  The base board of MicroAutoBox II (DS1401-21 and later) provides an acceleration sensor. The measurement is supported by RTLib and RTI.

For further information, refer to Onboard Sensors (MicroAutoBox Features).

**Pressure sensor**  Pressure measurement is now also supported by RTI.

For further information, refer to Onboard Sensors (MicroAutoBox Features).

**Watchdog handling**  There is new flexible watchdog handling for MicroAutoBox II (DS1401-23 and later) supported by RTLib. With a watchdog, you can monitor a task in your real-time application.

For further information, refer to Watchdog Handling (MicroAutoBox Features).

**MicroAutoBox configuration tool**  The `ds1401config` command line utility is now a dialog-based configuration tool, which you can open by calling `ds1401_config`. For easy access, you can integrate the tool as a user function, for example, in ControlDesk Next Generation.

For further information, refer to Integrating into a Network (MicroAutoBox II Hardware Installation and Configuration).

**Ejecting the USB Flight Recorder device**  For a safe removal of the USB mass storage device used for flight recording, there are now RTLib functions and an RTI block for ejecting the device.

For further information, refer to USB Flight Recorder (MicroAutoBox Features).
Loading slave applications to DS2302 boards

To load slave applications to a DS2302 board, RTI provides the new DS2302_DSP_SETUP_Bx block.

Slave applications are written in the C programming language. To load them to the DS2302, they must be converted to an intermediate format (SLC format). Then the block includes them in the real-time application for the processor board and loads them to the DS2302 DSPs during application start.

For further information, refer to DS2302_DSP_SETUP_Bx (DS2302 RTI Reference).

Migration Aspects of RTI/RTI-MP and RTLib

Loading slave application to DS2302 boards

As the concept for loading a slave application to DS2302 boards has changed (see New Features of RTI/RTI-MP and RTLib on page 101), you must modify your real-time model to use the new loading concept.

If the real-time model is implemented in Simulink, you must generate the SLC files for the slave applications, add the DS2302_DSP_SETUP_Bx block to your Simulink model and rebuild the real-time application. For details, refer to How to Load Slave Applications via RTI (DS2302 DSP Programming).

If the real-time model is hand-coded in C, you must generate the SLC files for the slave applications, add a loading function to your C program and rebuild the real-time application. For details, refer to How to Load a Slave Application via an RTLib Function (DS2302 DSP Programming).
## RTI Bypass Blockset

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of the RTI Bypass Blockset 2.9.1</td>
<td>105</td>
</tr>
<tr>
<td>Migrating to RTI Bypass Blockset 2.9.1</td>
<td>107</td>
</tr>
</tbody>
</table>

### New Features of the RTI Bypass Blockset 2.9.1

<table>
<thead>
<tr>
<th>RTI Bypass Blockset</th>
<th>Enhancements to the RTIBYPASS_FUNCTION_BLx block</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTIBYPASS_FUNCTION_BLx block</td>
<td>The RTIBYPASS_FUNCTION_BLx block provides the following enhancements:</td>
</tr>
<tr>
<td></td>
<td>■ The Variables page now makes it easier to integrate custom C code into the bypass application. When you add a source file, all its functions are parsed and displayed hierarchically with their parameters for you to select from.</td>
</tr>
<tr>
<td></td>
<td>■ The RTIBYPASS_FUNCTION_BLx block now supports global variables related to the source code files. When you add a source file containing information on global variables, the global variables are displayed on the Variables page. You can select these variables to be used as input and/or output values for a function call.</td>
</tr>
<tr>
<td></td>
<td>The input and output ports of a Function block are therefore now assigned by evaluating the parameters of the C function selected for the block, the function's return values, and the selected global variables.</td>
</tr>
</tbody>
</table>
The RTIBYPASS_FUNCTION_BLx block now supports value conversion. You can add an A2L file associated to an imported C module to the Function block. The conversion information for the function's input and output variables is imported to the model from the added A2L file. If value conversion is activated for a variable, the converted (physical) value is used for the variable in model context.

Refer to RTIBYPASS_FUNCTION_BLx (RTI Bypass Blockset Reference).

Reusing existing ECU variables with the internal bypass model
The RTI Bypass Blockset now allows you to reuse ECU variables that were already defined in the original ECU code and its associated A2L file with the internal bypass model. If enabled, the ECU application and the internal bypass application use the same signal and parameter instances, that is, they share the same memory locations. Otherwise, the internal bypass code declares its own representations. Reusing ECU variables saves ECU flash memory, since no extra memory has to be allocated.

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

Support of ASAM MCD-2 MC Ver. 1.6
The RTI Bypass Blockset supports A2L and AML files that are compatible with version 1.6 of the ASAM MCD-2 MC standard.

Support of enhancements to RTIBYPASS_FUNCTION_BLx block
The RTI Bypass Blockset MATLAB API supports the enhancements to the RTIBYPASS_FUNCTION_BLx block. It comes with new and modified methods for function handling.

For details, refer to the RTI Bypass Blockset MATLAB API Reference.

Method discontinuation
The following methods are obsolete due to changes in the RTIBYPASS_FUNCTION_BLx block:

- SET_HEADER
- GET_HEADER
- UNSET_HEADER
## Migrating to RTI Bypass Blockset 2.9.1

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

The current release comes with RTI Bypass Blockset 2.9.1, which is compatible with earlier blockset versions 2.x. However, data management has changed compared to RTI Bypass Blockset Version 2.5 or earlier.

If you have a Simulink model built with blockset version 2.5 or earlier and open it with RTI Bypass Blockset 2.9.1, the old Data Dictionary file (file name extension DD) is replaced by a new Data Dictionary file (file name extension VDB) using the information stored in the Setup block as soon as you open and close the Setup block dialog by clicking 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 2.9.1 and want to use it with RTI Bypass Blockset 2.5 or earlier, the model’s Data Dictionary file for the earlier blockset version (file name extension DD) is recreated 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 2.9.1 (file name extension VDB) remains on disk.

To enable the RTI Bypass Blockset to recreate the data dictionary, the ASAM-MCD 2MC (A2L) files specified in the Setup block must be accessible at the specified location and must be unchanged.

### Reusing a model with Function block in earlier RTI Bypass Blockset versions

The Function block in RTI Bypass Blockset 2.9.1 is not downward compatible with the Function blocks in earlier RTI Bypass Blockset versions. If you have a model with a Function block that was saved with RTI Bypass Blockset 2.9.1 and you want to use it with an earlier RTI Bypass Blockset version, you must delete the Function block and replace it by a new one from the earlier blockset version.

### Migration aspects of RTI Bypass Blockset MATLAB API

Due to changes in the RTIBYPASS_FUNCTION_BLx block, the RTI Bypass Blockset MATLAB API for RTI Bypass Blockset 2.9.1 is incompatible with earlier versions. You possibly cannot reuse existing batch scripts to automate the creation and configuration of RTI MATLAB models if they contain methods for function handling.
RTI CAN MultiMessage Blockset

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of the RTI CAN MultiMessage Blockset 2.7 109</td>
</tr>
<tr>
<td>Migrating to RTI CAN MultiMessage Blockset 2.7 110</td>
</tr>
</tbody>
</table>

New Features of the RTI CAN MultiMessage Blockset 2.7

Support of AUTOSAR System Template 3.2.2

The RTI CAN MultiMessage Blockset now also supports the AUTOSAR System Template based on AUTOSAR Release 3.2.2 for describing CAN networks.

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

Migrating to RTI CAN MultiMessage Blockset 2.7

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 Reference).

Reusing CAN bus configurations last saved in ControlDesk 3.7.2 or earlier

Suppose you worked in ControlDesk 3.7.2 or earlier with a CAN bus configuration based on a real-time application built with RTI CAN MultiMessage Blockset 2.5.4 or earlier. When you reuse this CAN bus configuration in ControlDesk 3.7.4 after rebuilding the application with RTI CAN MultiMessage Blockset 2.7, you cannot perform monitoring, logging and replay (due to a changed build process).

There are two ways to perform monitoring, logging and replay:

- Use the `DSMigrateBusCfg.exe` command line tool in ControlDesk 3.7.4 to migrate your existing CAN bus configuration to ControlDesk 3.7.4. Refer to Migrating to ControlDesk 3.7.4 on page 58.
- Create a new bus configuration in ControlDesk 3.7.4, and then add the required monitor, logger and replay nodes to the Bus Navigator tree again.
RTI LIN MultiMessage Blockset

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of the RTI LIN MultiMessage Blockset 2.0</td>
<td>111</td>
</tr>
<tr>
<td>Migrating to RTI LIN MultiMessage Blockset 2.0</td>
<td>112</td>
</tr>
</tbody>
</table>

New Features of the RTI LIN MultiMessage Blockset 2.0

Support of AUTOSAR System Template 3.2.2

The RTI LIN MultiMessage Blockset now also supports the AUTOSAR System Template based on AUTOSAR Release 3.2.2 for describing LIN networks.

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

Support of LDF files according to the SAE J2602 standard

The RTI LIN MultiMessage Blockset now also supports LDF files according to the SAE J2602 standard for describing LIN networks. You can base your configuration on an imported J2602-compliant LDF file.
Migrating to RTI LIN MultiMessage Blockset 2.0

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 RTI LIN MM blocks and save the model before modifying the LIN configuration.

To create new S-functions for all the RTI LIN MM 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
  
  ```matlab
  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 RTI LIN MM GeneralSetup block.

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

Reusing LIN bus configurations last saved in ControlDesk 3.7.2 or earlier

Suppose you worked in ControlDesk 3.7.2 or earlier with a LIN bus configuration based on a real-time application built with RTI LIN MultiMessage Blockset 1.8.2 or earlier. When you reuse this LIN bus configuration in ControlDesk 3.7.4 after rebuilding the application with RTI LIN MultiMessage Blockset 2.0, you cannot perform monitoring, logging and replay (due to a changed build process).

There are two ways to perform monitoring, logging and replay:

- Use the DSMigrateBusCfg.exe command line tool in ControlDesk 3.7.4 to migrate your existing LIN bus configuration to ControlDesk 3.7.4. Refer to Migrating to ControlDesk 3.7.4 on page 58.

- Create a new bus configuration in ControlDesk 3.7.4, and then add the required monitor, logger and replay nodes to the Bus Navigator tree again.
RTI FPGA Programming Blockset

Limited availability outside Europe and Asia, please inquire.

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of the RTI FPGA Programming Blockset 2.4</td>
</tr>
<tr>
<td>Migrating to RTI FPGA Programming Blockset 2.4</td>
</tr>
</tbody>
</table>

New Features of the RTI FPGA Programming Blockset 2.4

Enhancements to the RTI blocks of the Processor Interface library

In the PROC_SETUP_BL block, you can now also add FPGA model INI files to the build process in the Processor-Build model mode. FPGA subsystems can still only be added in FPGA-Build/Offline simulation model mode.

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

Enhancements to the RTI blocks of the FPGA Interface library

In the FPGA_SETUP_BL block, the dialog settings have been prepared for a future enhancement. The settings are currently deactivated.

For further information, refer to RTI FPGA Programming Blockset - FPGA Interface Reference.
The RTI FPGA Programming Blockset now supports Versions 13.4 and 14.1 of the Xilinx design tools.

<table>
<thead>
<tr>
<th>Xilinx Design Tools Version</th>
<th>Operating System</th>
<th>MATLAB Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.4&lt;sup&gt;1)&lt;/sup&gt;</td>
<td>▪ Windows XP Professional SP3 (32-bit version)</td>
<td>▪ MATLAB R2010bSP2&lt;sup&gt;2)&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>▪ Windows 7 Business, Ultimate, and Enterprise SP1 (32-bit version and 64-bit</td>
<td>▪ MATLAB R2011a</td>
</tr>
<tr>
<td></td>
<td>version)</td>
<td>▪ MATLAB R2011b</td>
</tr>
<tr>
<td>14.1&lt;sup&gt;1)&lt;/sup&gt;</td>
<td>▪ Windows XP Professional SP3 (32-bit version)</td>
<td>▪ MATLAB R2011a</td>
</tr>
<tr>
<td></td>
<td>▪ Windows 7 Business, Ultimate, and Enterprise SP1 (32-bit version and 64-bit</td>
<td>▪ MATLAB R2011b</td>
</tr>
<tr>
<td></td>
<td>version)</td>
<td></td>
</tr>
</tbody>
</table>

<sup>1)</sup> Only 32-bit version
<sup>2)</sup> Tested by dSPACE but not officially supported by Xilinx.

The DS5203 FPGA Board (LX50) also supports the WebPACK Editions of the Xilinx design tools.

Enhanced performance

- The startup phase of the build process is now up to eight times faster.
- The startup phase is the time between starting the build process and starting the ISE.
- Changing a parameter in a block dialog of an RTI FPGA Interface block is now up to six times faster.

Enhanced handcode support

The handcode interface now supports Verilog in addition to VHDL.

Simplified migration for DS5203 and DS5203M1 frameworks

If you use one of the DS5203 frameworks and want to switch to the DS5203M1 framework, you now do not need to manually replace all the blocks in your model that come from the RTI FPGA Programming Blockset. The framework migrate mechanism analyzes the blocks and automatically reparameterizes all the blocks that are compatible with the previous ones. After migration you get a list of the incompatible blocks, which you have to migrate manually.

This migrate mechanism also works for all the other frameworks, but there are no or only few compatible compliances.
For further information, refer to Using Different Frameworks (RTI FPGA Programming Blockset Guide).

**Related topics**

- Basics
- Migrating to RTI FPGA Programming Blockset 2.4 on page 115

---

**Migrating to RTI FPGA Programming Blockset 2.4**

**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 2.4**

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 2.4**

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 2.4, 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 115.

You also have to update the framework if you have updated from MATLAB R2008b or earlier to MATLAB R2009a or later.

**Updating the FPGA framework using the script interface**

Before you start migration, you should make a backup of your model.

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 subsystems 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')
SystemDesk 3.x

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of SystemDesk 3.2</td>
<td>118</td>
</tr>
<tr>
<td>Migrating to SystemDesk 3.2</td>
<td>127</td>
</tr>
</tbody>
</table>
New Features of SystemDesk 3.2

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>New General Features</th>
<th>118</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring and Generating V-ECUs</td>
<td>119</td>
</tr>
<tr>
<td>Simulating Systems</td>
<td>126</td>
</tr>
</tbody>
</table>

New General Features

SystemDesk 3.2 has the following new general features.

AUTOSAR Releases supported by SystemDesk 3.2

SystemDesk supports the following AUTOSAR Releases and Revisions:

<table>
<thead>
<tr>
<th>AUTOSAR Release</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>3.2.2 1)</td>
</tr>
<tr>
<td></td>
<td>3.2.1</td>
</tr>
<tr>
<td></td>
<td>3.1.5</td>
</tr>
<tr>
<td>3.1</td>
<td>3.1.4.DAI.4</td>
</tr>
<tr>
<td></td>
<td>3.1.4</td>
</tr>
<tr>
<td></td>
<td>3.1.2</td>
</tr>
<tr>
<td></td>
<td>3.1.0</td>
</tr>
<tr>
<td>3.0</td>
<td>3.0.7</td>
</tr>
<tr>
<td></td>
<td>3.0.6</td>
</tr>
<tr>
<td></td>
<td>3.0.4</td>
</tr>
<tr>
<td></td>
<td>3.0.2</td>
</tr>
</tbody>
</table>

1) New in SystemDesk 3.2
Configuring and Generating V-ECUs

RTE interventions for testing software components

SystemDesk lets you configure RTE interventions to test software components during a simulation run. RTE interventions are additional insertions to the original RTE code that let you access the communication of software components (SWCs) during a simulation run. You can use RTE interventions, for example, for stimulation or error injection to perform software component tests.

With RTE interventions (RTE intervention points and services), you can overwrite the original communication of software components during run time. You can connect internal or external stimuli to the software component ports via RTE interventions. Internal stimuli, for example, deriving from an extra stimulus SWC integrated on the V-ECU, are connected to the SWC to be tested via extra RTE service ports. External stimuli, for example, deriving from an environment model running on an connected environment VPU, are transmitted by data access points.
The illustration below shows you a simplified signal flow when using RTE interventions.

**RTE intervention mechanism**  To perform software component tests, SystemDesk lets you create RTE interventions for:
- Read/write access of data elements of unconnected ports of software components
- Read/write access of data elements of connected ports of software components
- Read/write access of interrunnable variables of software components
- Status return values of RTE API functions
RTE interventions are realized by using RTE intervention points and RTE intervention services.

- **RTE intervention points** are additional access points in the RTE code. They are generated only when you enable RTE interventions. RTE intervention points are called up during run time, for example, when a data element is read. For further processing, the RTE intervention point transmits data element read to an RTE intervention service that is linked to the RTE intervention point.

- **RTE intervention services** are C functions also generated in addition to the genuine RTE code. Each RTE intervention service is triggered by the RTE intervention point linked to it. The RTE intervention service specifies how to handle the data element value, operation argument or status return value that it gets form the RTE intervention point. The values/arguments can be forwarded and/or changed. For this purpose, an RTE intervention service can have up to two input signals (input and enable) and one output signal.

The illustration below shows the RTE intervention mechanism when working with sender/receiver interfaces.

In addition to what is said, SystemDesk generates A2L variables for the input and output signals of RTE intervention services. So you can access the RTE interventions also via these A2L variables with an experimenting software when executing the simulation system on VEOS.
Specifying RTE interventions  You can specify RTE interventions via an editor that provides access to the signals of an ECU. You can map one or more RTE intervention points to an RTE intervention service. So you can modify multiple RTE signals in the same way and at the same time.

The illustration below shows the an example in SystemDesk’s RTE interventions editor.

When generating the RTE code afterwards, SystemDesk instruments the RTE code, adds data access points or RTE service ports, and specifies A2L variables as specified.

Code example for an RTE intervention  The following listing shows a code example for an RTE intervention:

File Rte_<swc>.c

```c
Std_ReturnType Rte_Read_Plant_RpUpi_Upi(Float * Upi)
{
    Std_ReturnType Rte_Status = RTE_E_OK;
    /* read data element */
    *Upi = RTE_IP_READ(RTE_IP_0, float32, Rte_Signal_2);
    /* API call return */
    return RTE_IP_STATUS(RTE_IP_1, Rte_Status);
}
```


Support for the IO hardware abstraction module

You can now configure the IO hardware abstraction module and generate an implementation for it to be used with VEOS simulation. This allows you to map ECU signals to IO hardware abstraction ports.

The illustration above shows how you can select ECU signals that are to be accessible via the IO hardware abstraction module. First you select an interface from SystemDesk’s library. In a second step you select signals from the ECU that instantiate the preselected library interface.

When you generate a V-ECU implementation SystemDesk generates an IO hardware abstraction module with ports and an implementation. SystemDesk adds VPU ports during system build for accessing the ECU signals that you have mapped to IO hardware abstraction ports to be used with VEOS offline simulation.
Auto configure and generate V-ECU implementations

With this version SystemDesk supports the configuration and implementation of additional basic software modules such as IO hardware abstraction, ECU state manager, or BSW scheduler module.

SystemDesk can now automatically solve dependencies between basic software modules when you perform tasks such as generating RTE, COM stack, or V-ECU implementation. A V-ECU implementation is the implementation of a virtual ECU (model) that can be run in an offline simulation.
The illustration above shows the plugin methods that SystemDesk provides for the RTE module. When you generate a V-ECU implementation, you have to execute some of the plugin methods of the RTE module. First you have to update the RTE configuration, then generate the RTE component, and finally start RTE generation. Additionally it might be necessary to update other module configurations before updating the RTE configuration.

Exporting V-ECU implementations

SystemDesk now allows you to export V-ECU implementations. A V-ECU implementation is the implementation of the entire virtual ECU for the purpose of simulation. It is the starting point for offline simulation with VEOS as well as for real-time simulation on SCALEXIO (coming with a future release).

When you export a V-ECU implementation, SystemDesk gathers a collection of files that belongs to the V-ECU implementation in a CTLGZ file archive.
Files in a V-ECU implementation

Three kinds of files can be part of a V-ECU implementation:
- Configuration files (ARXML)
- Code files (H, C)
- A variable description file (A2L)

For details, refer to Export V-ECU Implementation (Container Management Document).

Simulating Systems

New features of simulating systems in conjunction with VEOS

**Offline simulation of ECU startup**
You can now download a simulation system without starting the simulation. This allows you to modify A2L variables before simulating systems.

This feature lets you simulate startup scenarios of V-ECUs.

**New stimulation scenarios**
Due to the new V-ECU generation features, SystemDesk 3.2 supports the following new stimulation scenarios in conjunction with VEOS:
- Modifying interrunnable variables and status return values of RTE API functions
- Stimulating an ECU or a bus breakdown
- Delaying the transmission of bus messages

The RTE interventions mechanism lets you stimulate:
- Data elements of software components
- Interrunnable variables and status return values of RTE API functions
- Operation arguments of client server calls
Migrating to SystemDesk 3.2

Using SystemDesk 3.1 projects with SystemDesk 3.2

SystemDesk 3.2 lets you open SystemDesk 3.1 projects. Each project is migrated automatically when opened the first time and can be saved as a SystemDesk 3.2 project.

To reuse in SystemDesk 3.2 a project last saved in SystemDesk 3.0 or earlier, you have to migrate it to SystemDesk 3.1 first. For detailed information on migrating projects to SystemDesk 3.1, refer to the related New Features and Migration document at http://www.dspace.com/goto?migration.

Simulating systems

With this version SystemDesk supports the configuration and implementation of additional basic software modules such as I/O hardware abstraction, ECU state manager, or BSW scheduler module. The ECU state manager (EcuM) configuration is mandatory for simulating systems with VEOS 3.0.

If you want to simulate the systems of SystemDesk 3.1 projects that you have migrated to SystemDesk 3.2 you have to perform the following steps:

1. Add the EcuM module configuration to each ECU configuration of the system you want to simulate.
   
   For instructions on adding module configurations, refer to How to Create Module Configurations (SystemDesk 3.x Guide).

2. Right-click the ECU configurations and select Auto Configure and Generate - V-ECU Implementation (SystemDesk 3.x Reference) to configure the ECU configuration and generate RTE code.

3. Build the simulation system. For instructions on building systems, refer to How to Build the Simulation System (SystemDesk 3.x Guide).
SystemDesk 4.x

New Features of SystemDesk 4.0

The new SystemDesk 4.0 supports the modeling of software and system architectures according to AUTOSAR 4.0.2 and 4.0.3. SystemDesk supports all the elements defined in the AUTOSAR schema. However, SystemDesk’s main focus is modeling the automotive software architecture, that is, the convenient design of software components with interfaces and internal behaviors.

Feature overview

- Modeling of AUTOSAR elements. SystemDesk 4.0 supports AUTOSAR 4.0.2 and 4.0.3 completely. Refer to Working with AUTOSAR Elements (SystemDesk 4.x Guide).
- Support of an AUTOSAR package view. Refer to Working with Packages (SystemDesk 4.x Guide).
- Assignment of AUTOSAR elements to ARXML files, facilitating the connection to version control systems. Refer to Basics on Assigning AR Elements to AUTOSAR Master Files (SystemDesk 4.x Guide).
Interoperability with other AUTOSAR-compliant tools:

- You can export software components from SystemDesk 4.0 for behavior modeling in TargetLink via container management. Refer to Integration of SystemDesk into an AUTOSAR Tool Chain ([SystemDesk 4.x Guide](#)).

- You can export a system extract from SystemDesk 4.0 for further use in basic software configuration tools. Refer to Create System Extract ([SystemDesk 4.x Guide](#)).

- Complete tool automation. Refer to Automating SystemDesk ([SystemDesk 4.x Guide](#)).

- Validation of elements in a SystemDesk project. Refer to Validating SystemDesk Elements ([SystemDesk 4.x Guide](#)).

### Migrating to SystemDesk 4.0

You cannot reuse or import SDP project files created with SystemDesk 3.x in SystemDesk 4.0, but you can reuse data via AUTOSAR conversion.

For details on migrating to SystemDesk 4.0 and on the different working concepts in SystemDesk 3.x compared with SystemDesk 4.0, refer to Migrating to SystemDesk 4.0 ([SystemDesk 4.x Guide](#)).
TargetLink

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New Features of TargetLink 3.4 and TargetLink Data Dictionary 3.4</strong></td>
<td>132</td>
</tr>
<tr>
<td><strong>Migrating to TargetLink 3.4 and TargetLink Data Dictionary 3.4</strong></td>
<td>158</td>
</tr>
</tbody>
</table>
New Features of TargetLink 3.4 and TargetLink Data Dictionary 3.4

For last-minute information on TargetLink 3.4 and on potential difficulties with model upgrades, it is recommended to visit the TargetLink 3.4 website at http://www.dspace.com/goto?TargetLinkDocumentationUpdate.

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Production Code Generation Features</td>
<td>132</td>
</tr>
<tr>
<td>New AUTOSAR-Related Features</td>
<td>146</td>
</tr>
<tr>
<td>New TargetLink Data Dictionary Features</td>
<td>150</td>
</tr>
</tbody>
</table>

New Production Code Generation Features

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component-Based Development Using Abstract Interfaces</td>
<td>133</td>
</tr>
<tr>
<td>Improvement for Custom Code</td>
<td>134</td>
</tr>
<tr>
<td>Optional Deactivation of Compute-Through-Overflow</td>
<td>134</td>
</tr>
<tr>
<td>Generating Virtual ECUs for Virtual ECU Testing</td>
<td>135</td>
</tr>
<tr>
<td>Specifications of the Target Simulation Modules</td>
<td>136</td>
</tr>
<tr>
<td>Code Generator Options</td>
<td>138</td>
</tr>
<tr>
<td>New API Commands</td>
<td>142</td>
</tr>
<tr>
<td>New Hook Functions</td>
<td>142</td>
</tr>
<tr>
<td>Access Function Changes</td>
<td>143</td>
</tr>
<tr>
<td>Improvement of Code Efficiency</td>
<td>144</td>
</tr>
<tr>
<td>General Enhancements and Changes</td>
<td>145</td>
</tr>
</tbody>
</table>
Component-Based Development Using Abstract Interfaces

Separating functionality to software modules

Support of component-based development allows you to separate the overall functionality into smaller software modules, called modular units (→ Modular unit (TargetLink Advanced Practices Guide)), which can be developed and tested independently. A separated modular unit can be tested with full test coverage more easily. Testing with full coverage is often not possible if the functionality is integrated in a larger context and therefore not all the test vectors can easily be applied directly at the modular unit. Thus the testability of modular units is generally higher. Later you only have to test the interaction between the modular units in the integration model for covering the overall functionality. Modular units also help you solve problems with long code generation times and high memory consumption during code generation for very large models, and can easily be reused without modification across different projects and in different contexts (→ Integration model (TargetLink Advanced Practices Guide)). This is an advantage if you want to change certain aspects of modular unit implementation: e.g., when you make a parameter of a modular unit a constant variable in one ECU project and a calibratable variable in another project. TargetLink provides abstract interfaces to support such changes without modifying the modular unit itself. You can map abstract interfaces to concrete implementations independently and after production code has been generated and tested for the modular unit (and should therefore not be changed).

For further information, refer to

- Basics on Component-Based Development Using Abstract Interfaces (TargetLink Advanced Practices Guide)
- Demo model: REPLACEABLE_DATA_ITEMS (TargetLink Demo Models)
- Incremental code generation for AUTOSAR SWCs on page 147
Improvement for Custom Code

| Enhancing unsupported Simulink blocks to TargetLink | Any unsupported Simulink block type (including subsystems) can be enhanced to a TargetLink Custom Code block (type II). Type II means that no S-function is generated from the custom code template, but the original MIL simulation behavior of the enhanced Simulink block is kept. For details, refer to Custom Code for Unsupported Simulink Blocks (TargetLink Advanced Practices Guide). |

| Width-invariant custom code template (type II) | A width-invariant custom code template can be (re)used in multiple Custom Code block (type II) instances of varying width contexts. The width context relates to the block’s input and output signals, and its states. Width invariance is achieved by using width macros in the custom code template. The width macros are resolved to the actual instance-specific widths at code generation time. For details, refer to Width-Invariant Custom Code Template (TargetLink Advanced Practices Guide). |

| Demo model | TargetLink provides a demo model, refer to CUSTOM_ENHANCEMENT (TargetLink Demo Models). |

Optional Deactivation of Compute-Through-Overflow

| Controlling the use of CTO patterns | By default, TargetLink uses compute-through-overflow (CTO) code patterns to improve code efficiency. Optionally, you can configure TargetLink to: 
- Never implement CTO patterns
- Always apply CTO code patterns whenever an overflow can occur
- Optimized use CTO patterns to improve the code efficiency
This allows you to adapt TargetLink to your company’s coding style. |

New Fixed-Point Library header file | The new header file sumprot.h was added to the Fixed-Point Library, used by TargetLink to avoid certain 64-bit calculations when configured to never implement CTO patterns. |
For further information, refer to

- **Code Generator Options** on page 138
  (The new Code Generator options.)
- **Resulting production code changes in comparison to the code generated by previous TargetLink versions** (refer to **Code Changes** on page 168)
- **‘Compute-Through-Overflows’ Property of Two’s Complement Arithmetic** ([TargetLink Production Code Generation Guide](#))
- **Macros and Functions Provided by the Fixed-Point Library** ([TargetLink File Reference](#))

### Generating Virtual ECUs for Virtual ECU Testing

**V-ECU Generation**

TargetLink lets you generate a V-ECU for a TargetLink subsystem and use it in virtual ECU testing (VET). V-ECUs emulate real ECUs in offline simulation scenarios.

You can download the V-ECUs to VEOS, dSPACE’s offline simulator. This allows seamless integration into the dSPACE tool chain and lets you automate tests with AutomationDesk or run experiments in ControlDesk Next Generation. This allows early testing and validation throughout the ECU development process.

**V-ECU Manager**

TargetLink’s V-ECU Manager assists you in configuring and performing the V-ECU build process.

**API commands**

New API commands allow you to integrate TargetLink’s V-ECU generation capability in your workflow.

For further information, refer to

- **V-ECU Manager** ([TargetLink Tool and Utility Reference](#))
- **New API Commands** on page 142
- **Generating Virtual ECUs for Virtual ECU Testing** ([TargetLink Advanced Practices Guide](#))
## Specifications of the Target Simulation Modules

The following table shows the combinations of evaluation boards, microcontrollers, and compilers supported by TargetLink 3.4 (TargetLink abbreviations).

| Evaluation Board | Microcontroller Type | Compiler
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Freescale HCS12EVB2)</td>
<td>Freescale MC9S12DP256</td>
<td>Cosmic 4.7</td>
</tr>
<tr>
<td>Freescale HCS12EVB2)</td>
<td>Freescale MC9S12DP256</td>
<td>Cosmic 4.8</td>
</tr>
<tr>
<td>Freescale HCS12EVB2)</td>
<td>Freescale MC9S12DP256</td>
<td>Metrowerks CodeWarrior 3.1</td>
</tr>
<tr>
<td>Freescale HCS12EVB2)</td>
<td>Freescale MC9S12DP256</td>
<td>Metrowerks CodeWarrior 5.1</td>
</tr>
<tr>
<td>MCT HCS12 T-Board (DP256)2)</td>
<td>Freescale MC9S12DP256</td>
<td>Cosmic 4.7</td>
</tr>
<tr>
<td>MCT HCS12 T-Board (DP256)2)</td>
<td>Freescale MC9S12DP256</td>
<td>Cosmic 4.8</td>
</tr>
<tr>
<td>MCT HCS12 T-Board (DP256)2)</td>
<td>Freescale MC9S12DP256</td>
<td>Metrowerks CodeWarrior 3.1</td>
</tr>
<tr>
<td>MCT HCS12 T-Board (DP256)2)</td>
<td>Freescale MC9S12DP256</td>
<td>Metrowerks CodeWarrior 5.1</td>
</tr>
<tr>
<td>MCT HCS12 T-Board (DP512)</td>
<td>Freescale MC9S12DP512</td>
<td>Cosmic 4.7</td>
</tr>
<tr>
<td>MCT HCS12 T-Board (DP512)</td>
<td>Freescale MC9S12DP512</td>
<td>Cosmic 4.8</td>
</tr>
<tr>
<td>MCT HCS12 T-Board (DP512)</td>
<td>Freescale MC9S12DP512</td>
<td>Metrowerks CodeWarrior 3.1</td>
</tr>
<tr>
<td>MCT HCS12 T-Board (DP512)</td>
<td>Freescale MC9S12DP512</td>
<td>Metrowerks CodeWarrior 5.1</td>
</tr>
<tr>
<td>MCT S12X T-Board2)</td>
<td>Freescale MC9S12XDP512</td>
<td>Metrowerks CodeWarrior 8.3</td>
</tr>
<tr>
<td>MCT S12X T-Board2)</td>
<td>Freescale MC9S12XDP512</td>
<td>Metrowerks CodeWarrior 8.7</td>
</tr>
<tr>
<td>MCT S12X T-Board USB2)</td>
<td>Freescale MC9S12XDP512</td>
<td>Metrowerks CodeWarrior 8.3</td>
</tr>
<tr>
<td>MCT S12X T-Board USB2)</td>
<td>Freescale MC9S12XDP512</td>
<td>Metrowerks CodeWarrior 8.7</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1766</td>
<td>Infineon c167</td>
<td>TASKING C166ST10 Toolchain 8.6</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1766</td>
<td>Infineon c167</td>
<td>TASKING TriCore VX-Toolset 3.2</td>
</tr>
<tr>
<td>Evaluation Board</td>
<td>Microcontroller Type</td>
<td>Compiler¹</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>----------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1766</td>
<td>Infineon TC1766</td>
<td>No longer supported TASKING TriCore VX-Toolset 3.5</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1766</td>
<td>Infineon TC1766</td>
<td>TASKING TriCore VX-Toolset 4.0</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1766 20 MHz</td>
<td>Infineon TC1766</td>
<td>No longer supported TASKING TriCore VX-Toolset 3.5</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1766</td>
<td>Infineon TC1766</td>
<td>TASKING TriCore VX-Toolset 4.0</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1767</td>
<td>Infineon TC1767</td>
<td>No longer supported TASKING TriCore VX-Toolset 3.5</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1767</td>
<td>Infineon TC1767</td>
<td>TASKING TriCore VX-Toolset 4.0</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1766 20 MHz</td>
<td>Infineon TC1766</td>
<td>TASKING TriCore VX-Toolset 4.0</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1796</td>
<td>Infineon TC1796</td>
<td>No longer supported TASKING TriCore VX-Toolset 3.5</td>
</tr>
<tr>
<td>Infineon TriBoard TriCore 1796</td>
<td>Infineon TC1796</td>
<td>TASKING TriCore VX-Toolset 4.0</td>
</tr>
<tr>
<td>Infineon EasyKit XC2287</td>
<td>Infineon XC2287</td>
<td>Gaio 11</td>
</tr>
<tr>
<td>Renesas M3A-2154</td>
<td>Renesas M32192</td>
<td>Gaio 9</td>
</tr>
<tr>
<td>Renesas M3A-2154</td>
<td>Renesas M32192</td>
<td>Renesas 5.1</td>
</tr>
<tr>
<td>Renesas EVB7058</td>
<td>Renesas SH-2E/SH7058</td>
<td>Renesas 9.4</td>
</tr>
<tr>
<td>Renesas EVB7058</td>
<td>Renesas SH-2E/SH7058</td>
<td>Renesas 9.4</td>
</tr>
<tr>
<td>Renesas SH72513 System Development Kit</td>
<td>Renesas SH-2A-FPUSH72513</td>
<td>Renesas 9.4</td>
</tr>
<tr>
<td>NEC Fx3-CAN it!</td>
<td>NEC Fx3-CAN it!</td>
<td>Green Hills 2012.1</td>
</tr>
<tr>
<td>NEC Fx3-CAN it!</td>
<td>NEC Fx3-CAN it!</td>
<td>NEC 3.40</td>
</tr>
</tbody>
</table>

1) Compiler Suite Version Supported
2) The board is no longer distributed by dSPACE but is still supported for downward compatibility

For detailed information on the evaluation boards supported by TargetLink, refer to TargetLink Evaluation Board Hardware Reference.

For further PIL support combinations that are part of a valid Software Maintenance Service (SMS) contract, refer to dSPACE’s TargetLink PIL Support website.
## Code Generator Options

The following new Code Generator options are available with TargetLink 3.4:

<table>
<thead>
<tr>
<th>Description</th>
<th>Explanation</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowStructAssignments</td>
<td>Copy structs or sub structs of identical type via a single assignment instead of assignments of all struct (substruct) components.</td>
<td>The C language allows direct assignments of struct variables. This is equivalent to copying all struct members but can instead be implemented by a compiler as copying the storage area (e.g. via memcpy()). This Code Generator option gives TargetLink permission to replace component-wise assignments of structs or sub structs by the assignment of whole structs (sub structs). If the situation is not sufficiently clear, e.g., due to code optimization, TargetLink will generate the component-wise assignments instead. Example: Consider: typedef struct S_tag { T3 c; T4 d; } S; typedef struct T_tag { T1 a; T2 b; S c; }; T S1; T S2; ... S2.a = S1.a; S2.b = S1.b; S2.c.d = S1.c.d; S2.c.e = S1.c.e; Here, the copy may be written instead as S2.a = S1.a; S2.b = S1.b; S2.c = S1.c; or S2 = S1;</td>
</tr>
</tbody>
</table>
### AssumeOperationCallsHaveNoUnknownDataFlow

For optimization purposes, treat every AUTOSAR operation as if the Data Dictionary Operation object has the NoDataFlowWithOtherOperations property set (or if the underlying function has a function class with the SIDE_EFFECT_FREE Optimization property set).

In principle, the implementation of an interaction between AUTOSAR operation calls is unknown. For other functions, TargetLink allows a function’s behavior to be specified via the function class Optimization property:

- “There is no unknown data flow via global variables, apart from the interface defined in the model” (SIDE_EFFECT_FREE):
  - Accesses to global variables and function calls accessing global variables can be moved past a call of the respective function.
- “There are no internal states, and the number of function calls along an execution path can be changed” (MOVABLE):
  - The function call can be moved into a conditionally executed control flow branch or to the second operand of a logical AND or OR operation.

There is usually no way to specify a function class for operation calls, unless they are also server runnables. Instead, TargetLink offers the NoDataFlowWithOtherOperations and NoStatesOrSideEffects properties at the Data Dictionary Operation object. This option allows you to ensure that all operation calls can be treated as if the function class has the SIDE_EFFECT_FREE Optimization property set or the operation has the NoDataFlowWithOtherOperations property set, i.e., it overrides the NoDataFlowWithOtherOperations property of all operations. If the option is switched off, then the value of the NoDataFlowWithOtherOperations property of each operation is evaluated.

### ExploitComputeThroughOverflow

Specifies whether and when compute-through-overflow code is implemented.

TargetLink uses compute-through-overflow (CTO) code patterns to implement efficient additions and subtractions and especially to avoid 64-bit operation. Since CTO can introduce overflows by casting the operands of an addition/subtraction to an unsigned data type, code checking tools might emit warnings or errors for this kind of code.

The possible values:

- 1 = Never: No compute-through-overflow code pattern will be implemented.
- 2 = Optimized: Apply compute-through-overflow patterns to improve the code efficiency.
- 3 = Always: Apply compute-through-overflow style code patterns whenever an overflow can occur.

The default is 2 = Optimized.
Utilize Value Equality Signal Line Split

<table>
<thead>
<tr>
<th>Description</th>
<th>Explanation</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replace block output variables of blocks preceding signal line splits by a variable with identical value, e.g., the state variable of a subsequent Unit Delay block.</td>
<td>If there is a signal line split after block B's output and the block code for one of the blocks with inputs driven by B's output leads to a plain value copy via a direct assignment to a variable of the same data type, then this option allows TargetLink to eliminate B's block output variable and replace it by the variable on the left-hand side of the assignment. This situation usually occurs if the output of B drives a TargetLink Outport block and is also used inside the system or if the output of B drives a Unit Delay block in addition to subsequent calculations. TargetLink optimizes the code pattern per se, as it may be generated in other constellations as well. Example: Without this optimization, you can find Float64 Sa1_Switch; static Float64 X_Sa1_Unit_Delay1 = 0.; if (Sa1_InPort1 &gt;= 0.) { Sa1_Switch = Sa1_InPort; } else { Sa1_Switch = (Sa1_InPort2 + X_Sa1_Unit_Delay1) * 0.5; } Sa1_OutPort1 = (Float64) sin(Sa1_Switch); Sa1_OutPort = Sa1_Switch; X_Sa1_Unit_Delay1 = Sa1_Switch;</td>
<td>on</td>
</tr>
</tbody>
</table>
UtilizeValueEqualityStructFunctionArguments

If a function has a parameter of pointer-to-struct type that is either an input or an output parameter, e.g., specified via TargetLink Bus InPort or OutPort blocks, and if the argument that is passed is the address operator applied to a struct variable, then for (if the option is activated) TargetLink is allowed to eliminate this struct variable if a complete copy from (or to) another struct takes place before (or after) the function call.

Note that this is effectively an elimination of all struct components that have to be assigned beforehand, so TargetLink considers the non-struct-type struct components of the struct variable and all the (recursively) contained substruct components, the “leaf components”, and can only perform the optimization if these components are eligible for the elimination of intermediate variables. Specifically, this means that the struct components have to have a default variable class or a user variable class with the ERASABLE Optimization property set.

Examples:

- Without this optimization, you can find:
  ```
  struct tag_T S1;
  struct tag_T S2;
  ...
  S2.a = S1.a;
  S2.sub.b = S1.sub.b;
  S2.sub.c = S1.sub.c;
  S2.d = S1.d;
  /* Pointer-to-struct parameter only used for input */
  r = f(&S2);
  ```

- or, if struct assignments are requested via the AllowStructAssignments Code Generator option,
  ```
  struct tag_T S1;
  struct tag_T S2;
  ...
  S2 = S1;
  /* Pointer-to-struct parameter only used for input */
  r = f(&S2);
  ```

  The optimization can replace S2, i.e.:
  ```
  struct tag_T S1;
  ...
  /* Pointer-to-struct parameter only used for input */
  r = f(&S1);
  ```

- Conversely, the TargetLink optimization
  ```
  struct tag_T S3;
  struct tag_T S4;
  ...
  /* Pointer-to-struct parameter only used for output */
  g(in1, in2, &S3);
  S4 = S3;
  ```

  can replace S3 if it is certain that all leaf components are assigned a new value in g(), i.e.:
  ```
  struct tag_T S4;
  ...
  /* Pointer-to-struct parameter only used for output */
  g(in1, in2, &S4);
  ```

  TargetLink needs to gather information about the accesses to the struct components inside the respective functions. This also counts against the memory limit set by the SideEffectFreeAnalysisThreshold option.
For reference information on all Code Generator options, refer to Code Generator Options (TargetLink Block and Object Reference).

### Migration aspects of Code Generator options

- **Obsolete Code Generator options**
- **Changed Code Generator options**
- **Recommended compatibility options for new Code Generator options**
  
  Refer to Migration Aspects Regarding Code Generator Options on page 174 for more information.

### New API Commands

**V-ECU generation**

New API commands allow you to integrate TargetLink’s virtual ECU (V-ECU) generation capability in your workflow:

- `tl_generate_vecu_implementation`
  
  (To generate V-ECU implementations that comprises all files necessary for virtual ECU testing.)

- `tl_compile_vecu`
  
  (To compile V-ECU implementations.)

- `tl_build_vecu`
  
  (To generate and compile V-ECU implementations in one step.)

For further information, refer to

- Generating Virtual ECUs for Virtual ECU Testing on page 135.
- API Commands (TargetLink API Reference).

### New Hook Functions

**AUTOSAR**

Frame model generation/update

TargetLink provides new hook functions for the AUTOSAR use case. You can use these hook functions to customize the frame model generation/update process.

- `tl_pre_add_comspecblock_hook.m` (TargetLink File Reference)
- `tl_post_add_comspecblock_hook.m` (TargetLink File Reference)
Access Function Changes

TargetLink’s access function mechanism was significantly improved. The following improvements were made:

**Specifying access functions** Optionally, TargetLink allows you to relate access functions to different variable kinds (scalar, vector, structure) by specifying the `VariableKindSpec` property of DD `AccessFunction` objects.

This allows you to:
- Specify variable access functions for struct variables
- Specify separate or common access functions for different variable kinds within one access function template and thus also via one DD `VariableClass` object

**Propagation into structures** You can now configure TargetLink to propagate access function templates (AFTs) into structures by specifying the `PropagateStructToComponents` property of DD `/Pool/Templates/AccessFunctions/<AccessFunctionName>/Settings` objects.

This allows you to:
- Specify an AFT applying to the leaf struct components (scalar or vector components) at the root struct variable (`PropagateStructToComponents` property set to on).
- Access the root struct variable itself by access function (AF) (`PropagateStructToComponents` property set to off).

**Controlling use of auxiliary variables** When specifying access function templates, you can now control whether TargetLink uses auxiliary variables to reduce the number of access function calls.

The specification is made at the DD `AccessFunction` objects’s `CreateLocalValueCopy` property.

For further information, refer to:
- Relating Access Functions to Variable Kinds (TargetLink Advanced Practices Guide)
- Propagation into structures ([TargetLink Advanced Practices Guide]).
- Controlling the Use of Auxiliary Variables with Access Functions ([TargetLink Advanced Practices Guide]).

### Improvement of Code Efficiency

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stronger code optimization by default</td>
<td>The TargetLink default settings of the Code Generator options that relate to code optimization have changed slightly. Production code generation leads to code that is even more optimized and readable, which is possible without burdening memory consumption and execution time in most cases.</td>
</tr>
<tr>
<td>Assignment of structs</td>
<td>TargetLink allows the assignment of structs as a whole: <code>struct_B = struct_A;</code> For details, refer to Optimizing Struct Variables ([TargetLink Advanced Practices Guide]).</td>
</tr>
<tr>
<td>Elimination of struct function arguments</td>
<td>If a function has a parameter of pointer-to-struct type and certain conditions are met, then TargetLink can eliminate this struct variable if a complete copy from/to another struct takes place before/after the function call. For details, refer to Optimizing Struct Variables ([TargetLink Advanced Practices Guide]).</td>
</tr>
<tr>
<td>Optimization of block outputs followed by signal line splits</td>
<td>TargetLink can eliminate superfluous block output variables if a block output is followed by signal line splits. For details, refer to Eliminating of Temporary Variables ([TargetLink Production Code Generation Guide]).</td>
</tr>
<tr>
<td>Scope reduction of structures</td>
<td>TargetLink can now reduce the scope of structures down to local. Because this implies setting the lifetime of the structure to <code>auto</code>, this optimization greatly reduces RAM consumption. For further information, refer to:</td>
</tr>
<tr>
<td>Scope of structured variables, in Obsolete Limitations on page 177</td>
<td></td>
</tr>
<tr>
<td>Optimizing Variables via Property Values</td>
<td>([TargetLink Advanced Practices Guide])</td>
</tr>
<tr>
<td>Scope Reduction of Variables</td>
<td>([TargetLink Advanced Practices Guide])</td>
</tr>
</tbody>
</table>
## General Enhancements and Changes

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reduced memory consumption for logging</strong></td>
<td>Basically, the memory consumption for logged simulation data has been optimized for all simulation modes (MIL, SIL and PIL simulation modes). If necessary, you can further reduce the memory consumption in MIL simulation mode. For details, refer to Logging speed and memory consumption in MIL simulation mode (TargetLink Production Code Generation Guide).</td>
</tr>
<tr>
<td><strong>Variable bit shifts in Stateflow</strong></td>
<td>You can specify variable bit shifts in Stateflow. For example, the operation ( x &lt;&lt; n ) shifts the value of ( x ) left by ( n ) bits. Refer to Guidelines on Specifying Stateflow models (TargetLink Advanced Practices Guide).</td>
</tr>
<tr>
<td><strong>Signal specification for the Constant block</strong></td>
<td>You can specify scaling parameters (Type, LSB, Offset) even though the default variable class is specified. Other blocks can then inherit these specified properties from a preceding Constant block. For details on the relevant block setting, refer to Allow signal specification (TargetLink Block and Object Reference).</td>
</tr>
</tbody>
</table>
| **New Fixed-Point Library header file**      | The `sumprot.h` header file was added to the Fixed-Point Library to avoid certain 64-bit calculations when suppressing compute-through-overflow (CTO). For further information, refer to
  - Optional Deactivation of Compute-Through-Overflow on page 134
  - Macros and Functions Provided by the Fixed-Point Library (TargetLink File Reference). |
| **Support of Simulink SLX model format**      | TargetLink 3.4 supports the Simulink SLX model format.                                                                                                                                                       |
| **Obsolete limitations**                     | With TargetLink 3.4, several limitations have been removed. For details refer to Obsolete Limitations on page 177.                                                                                             |
New AUTOSAR-Related Features

Features of the TargetLink AUTOSAR Module

<table>
<thead>
<tr>
<th>Supported AUTOSAR Releases</th>
<th>AUTOSAR Release</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.0</td>
<td>4.0.3(^1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.0.2</td>
</tr>
<tr>
<td></td>
<td>3.2</td>
<td>3.2.2(^1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.2.1</td>
</tr>
<tr>
<td></td>
<td>3.1</td>
<td>3.1.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.1.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.1.0</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>3.0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.0.2</td>
</tr>
<tr>
<td></td>
<td>2.1</td>
<td>2.1.4</td>
</tr>
</tbody>
</table>

\(^1\) New in TargetLink 3.4

Specifying AUTOSAR release in the TargetLink Data Dictionary

TargetLink allows you to generate AUTOSAR-compliant code for both AUTOSAR releases 3.x/4.x.

You can specify which AUTOSAR release to use in the DD /Pool/AUTOSAR/Config object the TargetLink Data Dictionary.

For information on generating AUTOSAR-compliant code, refer to Generating AUTOSAR-Compliant Code (TargetLink AUTOSAR Modeling Guide).

Working with new AUTOSAR 4.x data type concept

AUTOSAR 4.x introduced the distinction between application data types (ADTs) and implementation data types (IDTs).

TargetLink provides two wizards to assist you in the creation of ADTs and IDTs to provide a smooth transition to the AUTOSAR 4.x use case.
You can use the ApplicationDataType Creation Wizard to create ADTs for existing typedefs (IDTs) and the ImplementationDataType Creation Wizard to create IDTs for existing ADTs.

For information on working with ADTs/IDTs, refer to Basics on Working with Implementation and Application Data Types (TargetLink AUTOSAR Modeling Guide).

### Incremental code generation for AUTOSAR SWCs

TargetLink provides several modeling features for component-based development. You can now also use these features for AUTOSAR models.

TargetLink allows you to place your AUTOSAR software components and runnables in separate code generation units (CGUs), i.e., referenced models or subsystems configured for incremental code generation.

**Possible use cases:** TargetLink 3.4 supports the following cases:

<table>
<thead>
<tr>
<th>Feature: CGU Contains…</th>
<th>Purpose:</th>
<th>CGU Nesting: Can Contain CGUs of…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type A: Functional part of runnable</td>
<td>To separate a functional part of a runnable in one CGU. Must not contain AUTOSAR blocks. Modeling of calibratable parameters and per instance memories is possible.</td>
<td>Type A</td>
</tr>
<tr>
<td>Type A: Complete runnable</td>
<td>To separate a complete runnable in one CGU. Can contain AUTOSAR blocks.</td>
<td>Type A</td>
</tr>
<tr>
<td>Type C: SWC containing at least one runnable</td>
<td>To separate a complete SWC in one CGU. Runnables have to be modeled in Simulink subsystems or TargetLink CGUs.</td>
<td>Type A</td>
</tr>
<tr>
<td>Type D: Several runnables</td>
<td>To separate several runnables in one CGU. The runnables can belong to different SWCs. The CGU does not need to contain all the runnables of one SWC. Runnables have to be modeled in Simulink subsystems or TargetLink CGUs.</td>
<td>Type A</td>
</tr>
</tbody>
</table>
**Example**  The following screenshot shows a runnable containing a referenced model of type B.

Note that the referenced model contains AUTOSAR blocks. The Runnable block holds the parameters of the referenced model.

---

**For further information, refer to**

- *Component-Based Development Using Abstract Interfaces* on page 133

---

**Improved frame model generation**  TargetLink’s frame model generation feature allows you to generate TargetLink models with AUTOSAR blocks for Runnables and their interfaces from AUTOSAR files or AUTOSAR data contained in the Data Dictionary.

This feature has been further improved:

**Update capability**  TargetLink’s frame model generation now has an update capability that facilitates your round trips between TargetLink and software architecture tools such as SystemDesk.

You can now update existing frame models according to AUTOSAR files imported to the Data Dictionary. This enables you to quickly adapt your model to changes in the AUTOSAR files when implementing the software component in TargetLink.

TargetLink allows you to generate or update frame models for selected software components or groups of software components.
Update report  When updating an existing model, TargetLink generates an HTML update report that gives you quick access to changed model parts via hyperlinks.

<table>
<thead>
<tr>
<th>SwcModel Update Report</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Software component</td>
<td>Updated TargetLink object</td>
<td>TargetLink block</td>
</tr>
<tr>
<td>Address Memory</td>
<td>InMemory</td>
<td>InMemory</td>
</tr>
<tr>
<td>Sensor</td>
<td>mxSensor</td>
<td>mxSensor</td>
</tr>
<tr>
<td>Turntable</td>
<td>mx_Turntable</td>
<td>mx_Turntable</td>
</tr>
<tr>
<td>Turntable</td>
<td>mx_Turntable</td>
<td>mx_Turntable</td>
</tr>
<tr>
<td>Turntable</td>
<td>mx_Turntable</td>
<td>mx_Turntable</td>
</tr>
</tbody>
</table>

TargetLink’s SwcModel Update Report informs you about updated blocks and whether user action is required.

Further improvements  TargetLink’s frame model generator now supports
- Application errors
- Communication specifications
- Mode access points

For further information, refer to
- New Hook Functions on page 142
(Customizing the frame model generation/update process via hook functions.)
- Generating/Updating a Frame Model from AUTOSAR Data

Delivery of software components as object code

TargetLink now allows you to deliver software components as object code.

This means you can better protect your intellectual property and to deliver tested components.

You can specify the delivery format at the DeliveryFormat property of DD SoftwareComponent objects.

Compatibility mode  By default, the software components delivered as object code are compliant to AUTOSAR compatibility mode, which results in functions with standardized names and data structures.

You can use an RTE generator from any vendor (provided that the RTE generator supports the compatibility mode).

This provides better interoperability with software architecture tools.
For further information, refer to


**Improved handling of SIL/PIL simulations**

The handling of SIL/PIL simulations for operation call subsystems has been improved.

TargetLink now generates simulation code if the Use this subsystem for SIL/PIL simulation checkbox is deactivated for all operation call subsystems contained in the model.

If the same operation call subsystem is instantiated multiple times, TargetLink chooses one instance.

**Improved Container Manager**

The Container Manager has been improved. For details, refer to New Features of the Container Manager 3.2 on page 55.

---

**New TargetLink Data Dictionary Features**

**Objective**

The TargetLink Data Dictionary 3.4 (DD) has the following new features, enhancements and changes:

**Where to go from here**

Information in this section

<table>
<thead>
<tr>
<th>Compare and Merge Features</th>
<th>151</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improvements for Data Dictionary File Handling</td>
<td>152</td>
</tr>
<tr>
<td>Further Improvements of the Data Dictionary Manager</td>
<td>155</td>
</tr>
<tr>
<td>New DD MATLAB API Commands</td>
<td>155</td>
</tr>
</tbody>
</table>
Compare and Merge Features

Comparing and merging DD objects in the DD Comparison pane of the DD Manager

If you need to compare and merge DD objects, the DD Manager automatically opens a Comparison pane when you select two DD objects for comparison. It contains a synchronized tree structure of the compared DD objects including their child objects. You can compare and merge selected DD objects in one DD workspace, DD objects in different DD workspaces, or whole DD workspaces. The differences in the compared DD objects are indicated by color codes. This makes it easier for you to manage even the most complex DD object trees.

Benefits

- You can compare Data Dictionary workspaces and objects efficiently (inter- and intra-workspace comparisons). For more information, refer to How to Compare DD Objects in the DD Comparison Pane (TargetLink Data Dictionary Basic Concepts Guide).
You can merge Data Dictionary subtrees, objects and properties efficiently using the synchronized tree view. For more information, refer to How to Merge DD Objects and Property Values in the DD Comparison Pane (TargetLink Data Dictionary Basic Concepts Guide).

You can customize the comparison, for example, by ignoring specific object properties. Refer to How to Customize the Comparison of DD Objects (TargetLink Data Dictionary Basic Concepts Guide).

You can generate a report on the comparison result (XML or HTML). In a comparison pane, right-click anywhere to generate a report for the current comparison. As an alternative, you can start a comparison in a DOS command prompt: Use the DsddComp.exe command. For more information, refer to How to Generate a Comparison Report (HTML/XML) in the DD Comparison Pane (TargetLink Data Dictionary Basic Concepts Guide).

Improvements for Data Dictionary File Handling

Simplified loading of partial Data Dictionary files by storing their original locations

The Open command can read metadata that is stored together with a partial DD file so that its content can easily be reloaded to its original location. You can also multiselect DD files and open them with this command.

If you open a DD file by using the Open command in the File menu of your TargetLink Data Dictionary Manager, the loading behavior is as follows:

- If the file is a complete Data Dictionary, the active DD workspace is filled with the content of the DD file.
- If the file is a partial Data Dictionary with metadata about its original position, the Data Dictionary Manager automatically reads the original subtree and loads the file contents to that position.
- If the file is a partial Data Dictionary without metadata about its original position, the Data Dictionary Manager loads the content to a valid position according to the data model. If this is not possible, it is copied to a /tmp subtree. You can then move the objects to valid positions manually.

Partial DD files saved with TargetLink versions earlier than TL 3.3 contain no metadata.
Simplified inclusion of partial DD files with the Point of Inclusion dialog

The new Point of Inclusion dialog makes it easier to handle partial DD files, which can be included at specific positions in the DD subtree. You can now create inclusion points for partial DD files directly by selecting an object tree and using the Point of Inclusion dialog.

Benefits of the new Point of Inclusion dialog:

- Allows you to work with simplified file specification by using relative paths and paths relative to the main DD.
- Provides you information on the file content, for example, its position in the DD subtree.

For more information on how to include partial DD files, refer to How to Include Partial Data Dictionary Files (TargetLink Data Dictionary Basic Concepts Guide).
Simplified partition of a DD project file by separating partial DD files with the Point of Inclusion dialog

If you open a DD project file in your TargetLink Data Dictionary Manager, you can separate specific subtrees and save them as included partial DD files.

![Point of Inclusion dialog](image)

For more information on how to separate subtrees and save them as included partial DD files, refer to How to Separate DD Subtrees and Save them as Included Partial DD Files (TargetLink Data Dictionary Basic Concepts Guide).
Further Improvements of the Data Dictionary Manager

Object Explorer

Adding properties (columns) from the Property Value List to the Object Explorer has been improved.

In addition to the existing methods for adding properties to the Object Explorer, the TargetLink Data Dictionary Manager now offers an easier way: You can add them by selecting a DD object in the Object Explorer and then selecting the new checkboxes in the Property Value List.

For more information on adding properties to the Object Explorer, refer to How to Specify Properties (Columns) Shown in the Object Explorer (TargetLink Data Dictionary Basic Concepts Guide)

Property Value List

Properties that cannot be set are now greyed out in the Property Value List. A tooltip informs you why the property cannot be set.

Loading DD files

You can now load the same DD file into different workspaces.

DD IncludeFileGroup objects

You can now group DD DDIncludeFile objects in DD DDIncludeFileGroup objects. Nesting is possible.

New DD MATLAB API Commands

The following new DD MATLAB API commands are available:

- 'Compare'
- 'CreateComparisonReport'
- 'Duplicate'
- 'FindPrevious'
- 'GetCodeFiles'

New Features and Migration  November 2012  155
'GetDDFiles'  
'GetFileAttributes'  
'GetPathToSLObject'  
'IsFileRoot'

'Compare'
dsd('Compare',<objectIdentifier1>,<objectIdentifier2>,[attributeName1,attributeValue1,...])

To compare two objects and to build a comparison tree.

'CreateComparisonReport'
dsd('CreateComparisonReport',<objectIdentifier>[,attributeName1,attributeValue1,...])

To create an XML file that contains the results of an object comparison. The specified object must be a comparison tree root object. If the specified XML file exists, it is overwritten.

'Duplicate'
dsd('Duplicate',<objectIdentifier>)

To duplicate a DD object. The result must comply with the Data Model.

'FindPrevious'
dsd('FindPrevious',<objectIdentifier>[,attributeName1,attributeValue1,...])

Looks for an object which matches specified criteria. This command is like the 'FindNext' command except that the DD tree is traversed in reverse order. Use this command to look for objects iteratively in a loop.

'GetCodeFiles'
dsd('GetCodeFiles',<objectIdentifier>)

To return names of generated files that are associated with an object that resides in the /Subsystems area after code generation.

'GetDDFiles'
dsd('GetDDFiles'[<DD_identifier>])

To return information about partial DD files that have been loaded into a workspace. These are files that have been loaded either by file inclusion or explicitly, for example, with the Load or the AutoLoad command.

'GetFileAttributes'
dsd('GetFileAttributes',<fileName>)

To retrieve attributes of a DD file.

dsd('GetPathToSLObject' [,...])
dsd('GetPathToSLObject',<objectIdentifier>[,attributeName1,attributeValue1,...])

New Features and Migration  November 2012
To get the path to the Simulink system or SF object associated with a DD object that resides in the /Subsystems area after code generation.

```
dsnd('IsFileRoot' [...])
```

To check if a DD object is a file root object. Note that DD root objects are always file roots.

**Related topics**

- Basics
  - dsddman (TargetLink Data Dictionary MATLAB API Reference)
Migrating to TargetLink 3.4 and TargetLink Data Dictionary 3.4

Upgrade process

To migrate from TargetLink 3.3 to TargetLink 3.4, only the TargetLink Data Dictionary needs to be upgraded. This is done by the DD update process which automatically starts when a DD file is opened.

To migrate libraries/models from older TargetLink Releases, you also have to perform the migration steps of the intervening TargetLink Releases. Refer to the previous TargetLink Migration Guides available on your DVD.

You can launch an upgrade manually by using the t1_upgrade API command.
For last-minute information on TargetLink 3.4 and on potential difficulties, you are recommended to visit the TargetLink 3.4 website at http://www.dspace.com/goto?TargetLinkDocumentationUpdate.

- When upgrading models and libraries, first upgrade models or libraries that themselves do not reference any other libraries, i.e., the blocks/subsystems they contain have no links to other libraries. Start with the bottom library and then upgrade the libraries above it in ascending order. You must not open the model or a referencing library until this is done.

For related information on upgrading libraries, refer to:
- How to Upgrade Libraries and Models from TargetLink 2.x to 3.x on page 167
- How to Prepare TargetLink User Libraries for Upgrade (TargetLink Production Code Generation Guide)
- tl_upgrade (TargetLink API Reference)

Projects that were created under TargetLink 1.3 or even earlier versions cannot be upgraded directly to TargetLink 3.4 and TargetLink Data Dictionary 3.4. You must first perform an upgrade to a TargetLink 2.x version before you can upgrade to TargetLink 3.4 and TargetLink Data Dictionary 3.4.

- Previous New Features and Migration documents are available via Internet and on the dSPACE DVD. You can download them from http://www.dspace.com/goto?migration or read them from the dSPACE DVD (see the \Doc\Print\PreviousReleases folder). The PDF files are called TLNewFeaturesAndMigrationVer.x.y.pdf, where x.y stands for the release number.

- For information on upgrading Data Dictionary files containing AUTOSAR objects, refer to AUTOSAR-Related Migration Aspects on page 176.

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discontinued Data Dictionary Features</td>
<td>160</td>
</tr>
<tr>
<td>Migrating to TargetLink Data Dictionary 3.4</td>
<td>160</td>
</tr>
</tbody>
</table>
Discontinued Data Dictionary Features

**dsdd_upgrade**

The `dsdd_upgrade` command is no longer available.

To upgrade a DD project file, use the `dsdd('Upgrade',<DD_Identifier>)` command (refer to Upgrade (TargetLink Data Dictionary MATLAB API Reference)), or the DD Manager.

Migrating to TargetLink Data Dictionary 3.4

**Discontinued documentation**

The dSPACE Data Dictionary Data Model Reference is no longer available. Refer to the TargetLink Data Dictionary MATLAB API Reference instead, or refer to the user assistance in the TargetLink Data Dictionary Manager.

**Upgrading Data Dictionary files**

To migrate from TargetLink 3.3 to TargetLink 3.4, only the TargetLink Data Dictionary files need to be upgraded. This is done by the DD update process which automatically starts when a DD file is opened. Upgrading TargetLink models and libraries is not necessary.
**Deletion of Subsystem and <Application> areas**

Due to the changes in the data model, the data in the Subsystem and <Application> areas, which resulted from code generation and build processes performed before the upgrade, is not consistent with new data models. Upgrading a DD project file therefore deletes the Subsystem and <Application> areas from the Data Dictionary. You are asked to confirm the deletion.

In batch mode, the subsystem node is deleted without query, i.e., only a message is issued. To create correct data in the Subsystem and <Application> areas after the upgrade, for example, data needed for generating ASAP2 files or AUTOSAR XML files, generate code for all the code generation units again.

**Method to upgrade Data Dictionary files**

TargetLink Data Dictionary 3.4 provides an upgrade process that automatically upgrades DD files (DD0 workspace) to version 3.4.

The upgrade process can be called in three ways:

- **Automatically opening an old TargetLink model**
  
  When you open a TargetLink model with an old (not upgraded) DD project file, TargetLink first runs the TargetLink Data Dictionary’s upgrade process.
Via the Tools menu in the Data Dictionary Manager
Manually via Tools - Upgrade current DD in the DD Manager.

- Via the Tools menu in the Data Dictionary Manager
- Manually via Tools - Upgrade current DD in the DD Manager.

The menu command is available only if the DD project file needs upgrading. Otherwise it is disabled (grayed out).

Data Dictionary API command
To call the upgrade process via the Data Dictionary's API, type `dsdd('Upgrade')` in the MATLAB Command Window.

If the upgrade was successful, this is indicated in the MATLAB Command Window as shown below:

```matlab
>> dsdd('Upgrade')
anal =
   0
```

The asterisk in the title bar of the Data Dictionary Manager indicates that the content of the Data Dictionary has changed.

Preconditions for upgrading Data Dictionary files
To ensure the DD file is upgraded correctly, the following preconditions must be met:

- There must be write permission for the DD file and the file must not be write-protected. If Data Dictionary Include files are used, there must be write access to all of them.
If Data Dictionary Include files are used, they must be saved after updating as well as the main DD file. This ensures that the Data Dictionary partitions into several files. To save the Include files correctly, you must first make the appropriate settings for them, for example, in the Point of Inclusion dialog.
How to Upgrade a Data Dictionary Without Include Files

Objective

If you open a TargetLink model with an old, nonupgraded DD project file, you have to upgrade the Data Dictionary file.

Method

To upgrade a Data Dictionary without Include 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 older DD version is involved.

2. Click Yes if no Include files are used in the Data Dictionary.
   If Include files are used, abort the upgrade process and refer to How to Upgrade a Data Dictionary With Include Files on page 165.
   The Delete generated objects dialog automatically opens if the Data Dictionary’s Subsystem area contains objects generated with a previous TargetLink version.

3. Click Yes.

4. Save the Data Dictionary.
   Saving the Data Dictionary (with write permission to the relevant DD file) completes the upgrade of the DD file.

Result

The next time you open the DD file, the upgrade dialog will not open because the DD file is up-to-date.
**How to Upgrade a Data Dictionary With Include Files**

**Objective**
If you open a TargetLink model with an old, nonupgraded Data Dictionary file, you have to upgrade the Data Dictionary file.

**Method**

1. **To upgrade a Data Dictionary with Include files**
   1. Open the model and the referenced TargetLink Data Dictionary, or type `dsdd('open',<DDFile>)` in the MATLAB Command Window.
      The Data Dictionary needs upgrading dialog automatically opens if an earlier DD version is involved.
   2. Select No in the upgrade dialog.
   3. Under /Config/DDIncludeFiles, set the AutoLoad and AutoSave properties for each Include file as shown below.
This ensures that after the Data Dictionary and the Include files have been upgraded, the upgraded Include files are saved when the Data Dictionary is saved. You can set these properties for a large number of Include files via the Object Explorer.

You can also use the Point of Inclusion dialog to set the include file properties.

4 Start the DD upgrade (including the included files) via Tools - Upgrade current DD in the DD Manager, or enter dsdd('Upgrade') in the MATLAB Command Window.

5 Save the Data Dictionary (with write permission to the relevant DD file). This completes the upgrade of the DD file itself as well as the included partial DD files.

Result

The next time you open the DD file, the upgrade dialog will not open because the DD file is up-to-date and so are the included partial DD files. After the files have been properly upgraded, you might want to restore the old settings for the Data Dictionary Include files.
# How to Upgrade Libraries and Models from TargetLink 2.x to 3.x

## Objective

Libraries and models are upgraded in a fixed sequence of steps.

When upgrading models and libraries, first upgrade models or libraries that themselves do not reference any other libraries, i.e., the blocks/subsystems they contain have no links to other libraries. Start with the bottom library and then upgrade the libraries above it in ascending order. **You must not open the model or a referencing library until this is done.**

## Method

**To upgrade libraries and models from TargetLink 2.x to 3.x**

1. In the MATLAB Command Window, type
   
   ```matlab
dadd_manage_project('open', '<name>.dd')
   ```
   to load the required and already upgraded DD project file. (One way to upgrade DD project files is to use the ```dadd('Upgrade', <DD_Identifier>1)``` command, refer to **Upgrade** (TargetLink Data Dictionary MATLAB API Reference)).

2. Type
   
   ```matlab
tl_upgrade('Model', '<Library>.mdl', 'UpgradeLibs', 'off', 'Force', 'on', <other properties as needed>)
   ```
   to upgrade single libraries.

   **If you set the Force option to on, the model <name> is treated as a TargetLink 2.x model. To upgrade a library that was built with TargetLink 2.x, this option must be set to on. Setting UpgradeLibs to off does not upgrade other libraries referenced by the updated library.**

3. Save the upgraded library file(s), for example, `Library.mdl`.

4. Repeat steps 2 and 3 for all other libraries.

5. Run `tl_upgrade()` for the model(s) or open the model(s). If you open the model(s), the upgrade is started automatically.

## Result

You have upgraded your libraries and finally your model.

## Related topics

- References
  - `tl_upgrade` (TargetLink API Reference)
Code Changes

The following tables list code changes that are due to the TargetLink ExploitComputeThroughOverflow Code Generator option setting. The production code pattern presented in the TargetLink 3.4 column differ with respect to the Code Generator setting’s value. Annotations are italicized.

![Diagram](https://via.placeholder.com/150)

The representation of a constant value in an operation using compute-through-overflow can be different from its representation in an operation without the instrumentation of overflows.

**Sum with operands sharing the output’s scaling and data type**

The following table shows the code changes for a Sum block taking as input two Int16s that have the same scaling and data type as the output.

<table>
<thead>
<tr>
<th>TargetLink 3.4</th>
<th>≤ TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExploitComputeThroughOverflow = Optimized</td>
<td>Optimized</td>
</tr>
<tr>
<td>Sa1_SimpleI16Sum = (Int16) (Sa1_I16In1 + Sa1_I16In2);</td>
<td>Sa1_SimpleI16Sum = (Int16) (Sa1_I16In1 + Sa1_I16In2);</td>
</tr>
<tr>
<td>No CTO casts are made, because analysis of context shows that no overflow can occur.</td>
<td>The CTO casts (Int16) are made to guarantee defined overflow behavior.</td>
</tr>
<tr>
<td>ExploitComputeThroughOverflow = Never</td>
<td>Never</td>
</tr>
<tr>
<td>Sa1_SimpleI16Sum = (Int16) (Sa1_I16In1 + Sa1_I16In2);</td>
<td>Sa1_SimpleI16Sum = (Int16) (Sa1_I16In1 + Sa1_I16In2);</td>
</tr>
<tr>
<td>No CTO casts are made, because CTO is forbidden by Code Generator option.</td>
<td>No CTO casts are made, because CTO is forbidden by Code Generator option.</td>
</tr>
<tr>
<td>ExploitComputeThroughOverflow = Always</td>
<td>Always</td>
</tr>
<tr>
<td>Sa1_SimpleI16Sum = (Int16) (Sa1_I16In1 + Sa1_I16In2);</td>
<td>Sa1_SimpleI16Sum = (Int16) (Sa1_I16In1 + Sa1_I16In2);</td>
</tr>
<tr>
<td>Same code pattern as in TargetLink 3.3</td>
<td>Same code pattern as in TargetLink 3.3</td>
</tr>
</tbody>
</table>
The following table shows the code changes for a Sum block taking two Int16s as input. The scaling of I16In3 differs from the output’s scaling, which implies a right shift operation:

<table>
<thead>
<tr>
<th>TargetLink 3.4</th>
<th>TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExploitThroughOverflow = Optimized</td>
<td>ExploitThroughOverflow = Never</td>
</tr>
<tr>
<td>Sa1_RShiftI16Sum = (Int16) (((Int16) (Sa1_I16In3 &gt;&gt; 1)) + Sa1_I16In4);</td>
<td>No CTO casts are made, because analysis of context shows that no overflow can occur. The second Int16 cast is the result cast of the right shift operation.</td>
</tr>
<tr>
<td>ExploitThroughOverflow = Never</td>
<td>ExploitThroughOverflow = Always</td>
</tr>
<tr>
<td>Sa1_RShiftI16Sum = (Int16) (((Int16) (Sa1_I16In3 &gt;&gt; 1)) + Sa1_I16In4);</td>
<td>No CTO casts are made. CTO is forbidden by Code Generator option. The second Int16 cast is the result cast of the right shift operation.</td>
</tr>
<tr>
<td>ExploitThroughOverflow = Always</td>
<td>Same code pattern as in TargetLink 3.3.</td>
</tr>
<tr>
<td>Sa1_RShiftI16Sum = (Int16) (((UInt16) (Int16) (Sa1_I16In3 &gt;&gt; 1)) + ((UInt16) Sa1_I16In4));</td>
<td>The CTO casts (UInt16) are made to guarantee defined overflow behavior. The second Int16 cast is the result cast of the right shift operation.</td>
</tr>
</tbody>
</table>

Same code pattern as in TargetLink 3.3.
The following table shows the code changes for a Sum block taking two Int16s as input. The scaling of I16In5 differs from the output's scaling, which implies a left shift operation:

<table>
<thead>
<tr>
<th>TargetLink 3.4</th>
<th>TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ExploitComputeThroughOverflow = Optimized</code></td>
<td><code>ExploitComputeThroughOverflow = Optimized</code></td>
</tr>
<tr>
<td><code>Sa1_LShiftI16Sum = (Int16) (((UInt16) (Sa1_I16In5 &lt;&lt; 1)) + ((UInt16) Sa1_I16In6));</code></td>
<td><code>Sa1_LShiftI16Sum = (Int16) (((UInt16) (Sa1_I16In5 &lt;&lt; 1)) + ((UInt16) Sa1_I16In6));</code></td>
</tr>
<tr>
<td>CTO casts are made, because analysis of context shows that the first input does not fit in a 16-bit integer data type. The Int16 cast is omitted because suboperations of additions and subtractions are given an unsigned type, if an overflow could occur (refer to Unsigned suboperations for addition/subtraction on page 172).</td>
<td>The CTO casts (UInt16) are made to guarantee defined overflow behavior. The second Int16 cast is the result cast of the left shift operation.</td>
</tr>
<tr>
<td><code>ExploitComputeThroughOverflow = Never</code></td>
<td><code>ExploitComputeThroughOverflow = Never</code></td>
</tr>
<tr>
<td><code>Sa1_LShiftI16Sum = (Int16) (((Int32) (((Int32) Sa1_I16In5) &lt;&lt; 1)) + ((Int32) Sa1_I16In6));</code></td>
<td><code>Sa1_LShiftI16Sum = (Int16) (((Int32) (((Int32) Sa1_I16In5) &lt;&lt; 1)) + ((Int32) Sa1_I16In6));</code></td>
</tr>
<tr>
<td>Because CTO is forbidden, calculations are performed in Int32 to avoid overflows.</td>
<td>Because CTO is forbidden, calculations are performed in Int32 to avoid overflows.</td>
</tr>
<tr>
<td><code>ExploitComputeThroughOverflow = Always</code></td>
<td><code>ExploitComputeThroughOverflow = Always</code></td>
</tr>
<tr>
<td><code>Sa1_LShiftI16Sum = (Int16) (((UInt16) (Sa1_I16In5 &lt;&lt; 1)) + ((UInt16) Sa1_I16In6));</code></td>
<td><code>Sa1_LShiftI16Sum = (Int16) (((UInt16) (Sa1_I16In5 &lt;&lt; 1)) + ((UInt16) Sa1_I16In6));</code></td>
</tr>
<tr>
<td>Same code pattern as in TargetLink 3.3.</td>
<td>Same code pattern as in TargetLink 3.3.</td>
</tr>
</tbody>
</table>

**Sum with three inputs without assign arithmetic for accumulations**

![Diagram of Sum block with three inputs](https://via.placeholder.com/150)
The following table shows the code changes for a Sum block taking three Int16s as input sharing the same scaling as the output. Note that the DoNotUseAssignArithmeticForAccumulation Code Generator option has been enabled to suppress assign arithmetic for accumulations:

<table>
<thead>
<tr>
<th>TargetLink 3.4</th>
<th>TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExploitComputeThroughOverflow = Optimized</td>
<td></td>
</tr>
<tr>
<td><code>Sa1_SimpleI16SumTwice = (Int16) (((UInt16) Sa1_I16In7) + (UInt16) Sa1_I16In8)) + (UInt16) Sa1_I16In9));</code></td>
<td></td>
</tr>
<tr>
<td>CTO casts are made, because analysis of context shows that the result of the first addition does not fit in a 16-bit integer data type. The second Int16 cast is omitted because suboperations of additions and subtractions are given an unsigned type if an overflow could occur (refer to Unsigned suboperations for addition/subtraction on page 172).</td>
<td></td>
</tr>
<tr>
<td>ExploitComputeThroughOverflow = Never</td>
<td></td>
</tr>
<tr>
<td><code>Sa1_SimpleI16SumTwice = (Int16) (((Int32) Sa1_I16In7) + ((Int32) Sa1_I16In8)) + ((Int32) Sa1_I16In9));</code></td>
<td></td>
</tr>
<tr>
<td>Because CTO is forbidden, calculations are performed in Int32 to avoid overflows.</td>
<td></td>
</tr>
<tr>
<td>ExploitComputeThroughOverflow = Always</td>
<td></td>
</tr>
<tr>
<td><code>Sa1_SimpleI16SumTwice = (((UInt16) Sa1_I16In7) + (UInt16) Sa1_I16In8)) + (UInt16) Sa1_I16In9));</code></td>
<td></td>
</tr>
<tr>
<td>Same code pattern as in TargetLink 3.3.</td>
<td></td>
</tr>
</tbody>
</table>

**Modified code pattern for addition/substraction**

If an addition or substraction meets the following conditions, the resulting code guarantees a defined overflow behavior.

- The result would allow a smaller data type than the data type specified.
- An operand does not fit this smaller data type.

**Code example** Suppose an addition is defined as follows:

```c
I32Out = I32In1 + I32In2
```

The first operand also had the user-defined limits $\max = 70000$ and $\min = 60000$, and the second had $\max = -50000$ and $\min = -70000$. Thus, the result would always fit Int16. However, both operands do not fit Int16 (overflow). In consequence, the operands are cast to UInt16 so that the overflow is defined.

The code would be generated as follows:

<table>
<thead>
<tr>
<th>TargetLink 3.4</th>
<th>TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>I32Out = (Int32) (Int16) (((UInt16) I32In1) + ((UInt16) I32In2));</code></td>
<td><code>I32Out = (Int32) (Int16) (((Int16) I32In1) + ((Int16) I32In2));</code></td>
</tr>
</tbody>
</table>
Unsigned suboperations for addition/substraction

Operations that are operands themselves in addition/substraction are assigned an unsigned data type, if they could overflow and if they apply Compute-Through-Overflow (refer to Sum with one operand differently scaled as output, implying a left shift on page 169).

**Code example**

Suppose an addition is defined as follows:

\[ I16Out = I16in1 + I16in2 \]

with LSB of \( I16Out \), \( I16in2 \) = \( 2^{-1} \), and LSB of \( I16in1 \) = \( 2^{0} \)

The code would be generated as follows:

<table>
<thead>
<tr>
<th>TargetLink 3.4</th>
<th>≤ TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I16Out = (\text{Int16}) (\text{UInt16}) (I16in1 &lt;&lt; 1) + (\text{UInt16}) I16in2 )</td>
<td>( I16Out = (\text{Int16}) (\text{UInt16}) (\text{Int16}) (I16in1 &lt;&lt; 1) + (\text{UInt16}) I16in2 )</td>
</tr>
</tbody>
</table>

Deletion of unused function parameters

The following applies to function parameters that are specified for chart functions and graphical functions and are propagated to all the subfunctions (Stateflow state and auxiliary functions):

Function parameters of these subfunctions that are not used are deleted although the Optimization property of their variable class in the TargetLink Data Dictionary is not set to ERASABLE.

Casts in Stateflow shift operations

Shift operations in Stateflow make use of additional casts, if the operand meets the following conditions:

- Operand uses a TargetLink data type different from the Stateflow data type.
- Operand is not scaled.

Casts of pointers (function calls)

With this version of TargetLink the cast behavior of pointers has changed as follows:

- If a function parameter has a type prefix that is interpreted as an address qualifier (e.g., _far), the type prefix is evaluated to check whether casts are necessary. In older TargetLink versions, a cast was applied for any type prefix.
- If a function parameter is a vector and the required and the actual type qualifiers differ, the actual type qualifier is cast to the required type qualifier. In older TargetLink versions, no such cast was applied.
Type guard identifiers (MISRA)

Type guard identifiers are now output without leading or trailing underscores, which is in accordance with MISRA. For example:

<table>
<thead>
<tr>
<th>TargetLink 3.4</th>
<th>≤ TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>#ifndef MyType_TYPE</td>
<td></td>
</tr>
<tr>
<td>#define MyType_TYPE</td>
<td></td>
</tr>
<tr>
<td>typedef MyType;</td>
<td></td>
</tr>
<tr>
<td>#endif /* MyType_TYPE */</td>
<td>#ifndef <em>MyType_TYPE</em></td>
</tr>
<tr>
<td>#define <em>MyType_TYPE</em></td>
<td></td>
</tr>
<tr>
<td>typedef MyType;</td>
<td></td>
</tr>
<tr>
<td>#endif /* <em>MyType_TYPE</em> */</td>
<td></td>
</tr>
</tbody>
</table>

Macro initialization (MISRA)

If a macro has a negative initialization value, the value is wrapped in parentheses. This is in accordance with MISRA. For example:

<table>
<thead>
<tr>
<th>TargetLink 3.4</th>
<th>≤ TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>#define MACRO ((Int8) -1)</td>
<td></td>
</tr>
<tr>
<td>#define SECOND_MACRO (-10)</td>
<td>#define MACRO (Int8) -1</td>
</tr>
<tr>
<td>#define SECOND_MACRO -10</td>
<td></td>
</tr>
</tbody>
</table>

&& and || operations (MISRA)

For function calls, access to vector components via [ ] operator, and access to the elements of structs, the following new rule applies, which is in accordance with MISRA: The arguments of && and || operations are wrapped in parentheses, for example:

<table>
<thead>
<tr>
<th>TargetLink 3.4</th>
<th>≤ TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A() &amp;&amp; b c</td>
<td></td>
</tr>
</tbody>
</table>

Relay block

For the TargetLink Relay block, the Switch On condition is now checked first to mimic the Simulink behavior in case the SwitchOn and SwitchOff values are equal.

Unit Delay block

If the block output has saturation enabled, a block output variable is always generated. Additionally, the assignment output = state is saturated.

In addition, the block output variable is always read by the succeeding block, if it meets one of the following conditions:

- Block output variable is logged.
- Block output variable is referenced from the TargetLink Data Dictionary.
- Block output variable has a user-defined variable class.
- Block output variable has saturation enabled.
Migration Aspects Regarding Code Generator Options

Basics on default changes
The settings of the Code Generator options are stored with the model (model-based option storage). In addition, you can store user-defined sets of Code Generator options in OptionSets in the TargetLink Data Dictionary since TargetLink version 3.1 (DD-based option storage). You can use DD-based option settings as a central source for overwriting model-based option settings.

If a model-based option value equals the default value of an older TargetLink version, it is changed to the new default value during upgrade. If a DD-based option value equaled the old default value, it is not changed to the new default value during upgrade but keeps the old value.

For details on both storage kinds, refer to Basics of Configuring the Code Generator for Production Code Generation (TargetLink Advanced Practices Guide).

Changed default values of Code Generator options
The default values of the Code Generator options listed below were changed:

<table>
<thead>
<tr>
<th>Code Generator Option</th>
<th>Default Value</th>
<th>≤ TargetLink 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowInterleavingCodeForAllSubsystems</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>ExploitRangesIndepentOfErasable</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>GlobalOptimizationIterationThreshold</td>
<td>30</td>
<td>10</td>
</tr>
<tr>
<td>OptimizationIterationThreshold</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>SideEffectFreeAnalysisThreshold</td>
<td>1000</td>
<td>10</td>
</tr>
</tbody>
</table>
Recommended compatibility settings for new Code Generator options

For the best downward compatibility to earlier TargetLink versions (≤ TargetLink 3.3), it is recommended to use the Compatibility Setting values for the new Code Generator options listed in the following table:

<table>
<thead>
<tr>
<th>Code Generator Option</th>
<th>Compatibility Setting</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowStructAssignments</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>AssumeOperationCallsHaveNoUnknownDataFlow</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>UtilizeValueEqualitySignalLineSplit</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>UtilizeValueEqualityStructFunctionArguments</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>ExploitComputeThroughOverflow</td>
<td>3 = Always</td>
<td>2 = Optimized</td>
</tr>
</tbody>
</table>

New Code Generator options

For more information on new Code Generator options, refer to Code Generator Options on page 138.

Related topics

- Code Generator Options (TargetLink Block and Object Reference)

Changes in TargetLink and TargetLink Data Dictionary API Functions

Obsolete DD MATLAB API commands

The following DD MATLAB API command is no longer supported:

- `dadd_upgrade`

  To upgrade a DD project file, use the `dadd('Upgrade', [DD_Identifier])` command (refer to `upgrade` (TargetLink Data Dictionary MATLAB API Reference)), or the DD Manager.
Various Migration Aspects

Style definition file cconfig.xml

For the %DSPACE_ROOT%\Matlab\TL\config\codegen\cconfig.xml style definition file, the <TL:generation-date ...> element was renamed. The element now reads as follows: <TL:generation-date-comment ...>.

If you use your own style definition file, you must adapt it.

The style definition file is referenced in the TargetLink Main Dialog block on the Advanced page.

AUTOSAR-Related Migration Aspects

Validation of AUTOSAR data

The validation of AUTOSAR data was significantly improved. In particular, badly/insufficiently specified ComSpecs are now detected. This improves data exchange with system architecture tools such as SystemDesk and third-party RTE generators.

AUTOSAR File Import/Export

The following two properties are obsolete:
- Merge
- EnablePackageSupport

The value of both properties is now always set to on. Adapt existing scripts accordingly.

RTE frame module

The default name of RTE frame modules has changed from Rte_SN to Rte.

It is no longer possible to generate additional variables or functions into the RTE frame module. Contact dSPACE Support concerning the migration of TargetLink models containing workarounds for RTE API functions unsupported in earlier TargetLink versions.
## Obsolete Limitations

With TargetLink 3.4, a couple of limitations existing in previous TargetLink versions have been removed. They are listed below.

### Component-Based Development Limitations

<table>
<thead>
<tr>
<th>AUTOSAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR blocks are not supported in referenced models or subsystems configured for incremental code generation.¹</td>
</tr>
</tbody>
</table>

¹ AUTOSAR blocks still are not allowed in model-based code generation units containing functional sub-parts of runnables.

### Code generation limitation

<table>
<thead>
<tr>
<th>Struct data types cannot be accessed by access functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access functions are not supported for structures as such. Only access to the components of a structure which are not themselves of struct type is supported.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scope of structured variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetLink reduces the scope of structures only to static local at most but not to local, even if this is technically possible. If the structure has the local scope to start with, it keeps the local scope.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Access function kind ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementing the ADDRESS access function kind in the form of macros would require different implementations for scalar and vector variables. Currently, only one implementation can be specified, and it is therefore necessary to use separate variable classes for scalar and vector variables that reference different AccessFunction templates.</td>
</tr>
</tbody>
</table>

### Stateflow limitations

<table>
<thead>
<tr>
<th>Shift by variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetLink does not support shift operations with a shift value that is a variable. For example:</td>
</tr>
<tr>
<td>- a &gt;&gt; 5; // ok</td>
</tr>
<tr>
<td>- a &gt;&gt; ShiftValue; // not supported</td>
</tr>
</tbody>
</table>
## AUTOSAR limitations

### RTE API calls in referenced models

TargetLink does not support the use of AUTOSAR communication in referenced models. If AUTOSAR data or parameters shall be used in referenced models, the data and parameters have to be
**VEOS**

**New Features of VEOS 3.0**

VEOS is the successor to the dSPACE Offline Simulator. The new features described below are enhancements and changes compared with dSPACE Offline Simulator 2.2p2.

dSPACE Release 7.4 provides VEOS as a new software product to perform offline simulation on your host PC. However, you must install an add-on to dSPACE Release 7.4 to work with VEOS in connection with one or more of the following dSPACE products:

- AutomationDesk
- ControlDesk Next Generation
- dSPACE HIL API (MAPort)
- ModelDesk
- Real-Time Testing (RTT)

New dSPACE Target for Offline Simulation features

**Inclusion of a dSPACE-specific measurement service**  When you generate an environment VPU using the dSPACE Target for Offline Simulation, a dSPACE-specific measurement service is included in the generated VPU. This allows you to perform triggered measurements in ControlDesk Next Generation, for example. For details on triggered measurements, refer to Configuring Triggered Measurement on dSPACE Platforms (ControlDesk Next Generation Basic Practices Guide).

**Real-Time Testing support**  When you generate an environment VPU using the dSPACE Target for Offline Simulation, you can now enable Real-Time Testing support. This allows you to use Real-Time Testing features, for example, to stimulate VPU variables with ControlDesk Next Generation’s Signal Editor in an offline simulation. For details on using the Signal Editor, refer to Using the Signal Editor (ControlDesk Next Generation Advanced Practices Guide).

Improved support for stimulation scenarios

SystemDesk 3.2 provides new methods for accessing and modifying variables in an offline simulation. In an offline simulation with VEOS, you can access and modify these variables for stimulation purposes.

In an offline simulation with VEOS, you can access these variables with ControlDesk Next Generation via entries in the A2L file.

**Stimulation scenarios involving ECU-internal variables**  VEOS supports the access to and modification of variables for the following stimulation scenarios involving ECU-internal variables:
- Data elements and operation arguments
- Interrunnable variables
- RTE status variables

**Stimulation scenarios involving an entire V-ECU**  VEOS supports the access to and modification of variables for the following stimulation scenarios involving an entire V-ECU:
- Stimulation of the KL 15 and KL 30 signals (ECU start-up scenario)
- Stimulation of the break down of bus communication (possible for each communication controller individually)
- Interruption of I/O signals
- Stimulation of VPU ports with an environment VPU
New VEOS Player

VEOS provides the *VEOS Player*. It lets you connect VPU ports in a simulation system, and configure and run an offline simulation.

The VEOS Player combines the functionality of the following software products which were part of the dSPACE Offline Simulator. The products are discontinued with VEOS 3.0.

- Offline Simulation Player
- Simulation System Editor

Refer to the *VEOS Player Document*. 
Compatibility Information

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>Supported MATLAB Releases</th>
<th>183</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>184</td>
</tr>
<tr>
<td>Run-Time-Compatibility of dSPACE Software</td>
<td>187</td>
</tr>
<tr>
<td>Limitations for Windows Vista/Windows 7</td>
<td>188</td>
</tr>
<tr>
<td>Limitations for 64-Bit Windows Operating Systems in Combination with dSPACE 32-Bit Software</td>
<td>189</td>
</tr>
</tbody>
</table>

Supported MATLAB Releases

The following MATLAB releases are supported by the dSPACE products on Release 7.4:

- R2012a, R2012b:
  - Both not supported by the RTI FPGA Programming Blockset - FPGA Interface
  - Problems using dSPACE Automotive Simulation Models (ASM): Due to performance problems in MATLAB R2012a and 2012b, it is recommended to install the following bugfix before you use ASMs with MATLAB R2012a or 2012b. Refer to http://www.mathworks.de/support/search_results.html?q=827771.

- R2011a, R2011b
- R2010b SP2 (tested with RTI FPGA Programming Blockset but not officially supported by Xilinx)
For up-to-date information on additional MATLAB releases which can be used in combination with dSPACE software, refer to http://www.dspace.com/goto?sw3rdparty.

## Operating System

<table>
<thead>
<tr>
<th>Operating system on host PC</th>
<th>The following operating systems are supported by the dSPACE products on Release 7.4:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Windows XP Professional with Service Pack 3 (32-bit version)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Windows XP Professional x64 Edition is not supported in general.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Windows Vista Business, Ultimate, and Enterprise with Service Pack 2 (32-bit version)</strong></td>
</tr>
<tr>
<td></td>
<td>Only the listed editions are supported. The Windows Vista Home and Starter editions are not supported.</td>
</tr>
<tr>
<td></td>
<td><strong>Windows 7 Professional, Ultimate, and Enterprise with Service Pack 1 (32-bit or 64-bit version)</strong></td>
</tr>
<tr>
<td></td>
<td>Only the listed editions are supported. The Windows 7 Home and Starter editions are not supported.</td>
</tr>
</tbody>
</table>
64-bit operating systems

The 64-bit Windows operating systems are supported by 32-bit dSPACE software running in WoW64 (Windows-On-Windows64). dSPACE software supports only the 64-bit version of Windows 7. Other 64-bit operating systems (Windows XP and Windows Vista) are not supported.

Some additional limitations apply when you use a 64-bit Windows operating system with dSPACE 32-bit software. Refer to Limitations for 64-Bit Windows Operating Systems in Combination with dSPACE 32-Bit Software on page 189.

Windows Vista, Windows 7:
Some general limitations apply when you use Windows Vista or Windows 7 in combination with dSPACE software. Refer to Limitations for Windows Vista/Windows 7 on page 188.

64-bit versions of TargetLink and Model Compare
The 64-bit versions of TargetLink and Model Compare require Windows 7 Professional, Ultimate, or Enterprise (64-bit version) with Service Pack 1. Other 64-bit operating systems (Windows XP and Windows Vista) are not supported.

Some additional limitations apply when you use the 64-bit version of TargetLink. Refer to Limitations for TargetLink 64-Bit Version (Quick Software Installation Guide).

ControlDesk Next Generation
- ControlDesk Next Generation’s ECU Diagnostics v2.0.1 device (supporting ASAM MCD-3 D V2.0.1) does not support Windows 7. As an alternative, you can use the ECU Diagnostics v2.0.2 device (supporting ASAM MCD-3 D V2.0.2), which supports Windows 7.
- ControlDesk Next Generation can also be installed on the MicroAutoBox Embedded PC, running under Microsoft Windows 7 Ultimate (32-bit version).
Compatibility Information

**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 that allows 32-bit Windows-based applications to run seamlessly on 64-bit versions of Windows. This allows you to use up to 4 GB virtual memory for each 32-bit process. 32-bit versions of Windows can address only up to 3.2 GB of memory in total for all running processes including the operating system itself.

**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:

The 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)
- Windows Server 2003 (32-bit or 64-bit version)
Run-Time-Compatibility of dSPACE Software

Definition

Run-time compatibility means:
- 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.

General notes on limitations

If dSPACE products interact directly (for example, through automation interfaces) or indirectly (for example, through common file types like .A2L limitations may apply. For major limitations, see below. For minor limitations, refer to the appropriate product documentation.

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 for the patch itself, refer to http://www.dspace.com/go/CompPatch.

Compatibility of products in dSPACE Release 7.4

Each software product in dSPACE Release 7.4 is run-time-compatible with the other software products in dSPACE Release 7.0 up to dSPACE Release 7.4. Note that RCP and HIL software products (on Release 7.4) cannot be used in combination with RCP and HIL software products from earlier dSPACE Releases.

Limitations for TargetLink and Model Compare

- With Model Compare 2.1 (on dSPACE Release 7.1) and earlier, you cannot dump TargetLink 3.4 models. If you need to dump TargetLink 3.4 models with Model Compare 2.1, contact dSPACE support for a compatibility patch.
- 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).
Compatibility Information

Limitation for ControlDesk Next Generation (= ControlDesk as of ControlDesk 4.0) When ControlDesk 4.3 is used with dSPACE Release 7.0 or earlier, limitations regarding Real-Time Testing (RTT) apply. Simulink models compiled with these releases are not supported by the Signal Editor (for signal generation) and for CAN Data Replay of the Bus Navigator. For details, refer to Problem with Incompatibility Between RTT 2.0 and 1.6 (Software Installation and Management Guide).

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.

Limitations for Windows Vista/Windows 7

Objective

Some limitations apply when you use Windows Vista/Windows 7 in combination with dSPACE software.

MATLAB support

For system requirements of MathWorks® software, refer to http://www.mathworks.com/support/sysreq/current_release.

Fast user switching not supported

The dSPACE software does not support the fast user switching feature of Windows.

Closing dSPACE software before PC shutdown

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.

User Account Control

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 to the required network drives.
### Compatibility Information

#### USB devices under Windows 7
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.

#### Using boards with ISA interface installed in the host PC
When installed directly in the host PC, ISA boards such as the DS1103 or DS813 cannot be used with Windows Vista/Windows 7 with the standard installation routines. If necessary, contact dSPACE Support.

---

## Limitations for 64-Bit Windows Operating Systems in Combination with dSPACE 32-Bit Software

### Objective
Some additional limitations apply when you use 64-bit versions of Windows 7 in combination with dSPACE 32-bit software.

### Limitations of device drivers
Third-party bus interfaces (CAN, LIN, or FlexRay) are supported only if they have 64-bit drivers from the manufacturers.

### Limitations of target compilers
For information on support for a specific target compiler, contact the respective compiler manufacturer.

### MATLAB
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. Click OK and continue installing the 32-bit version of MATLAB.
ASM blocksets
new features 32
ASM Diesel Engine Blockset
migrating 34
new features 33
ASM Diesel Exhaust Blockset
migrating 35
ASM Electric Components Blockset
migrating 36
ASM Engine Gasoline Basic Blockset
migrating 39
new features 38
ASM Environment Blockset
migrating 41
new features 40
ASM Optimizer Blockset
migrating 42
ASM Traffic Blockset
migrating 43
new features 43
ASM Truck Blockset
migrating 45
ASM Turbocharger Blockset
migrating 46
ASM Vehicle Dynamics Blockset
migrating 47
new features 47
AutomationDesk
migration 30
new features 37
ECU Interface Manager
migration 87
new features 85
general enhancements and changes 14
host PC software
operating system 184
key features 19
Limitations
TargetLink
obsolete limitations 177
dSPACE FlexRay Configuration Package
new features 79
dSPACE HIL API .NET
new features 77
dSPACE Offline Simulator
new features 179
dSPACE Python Extensions
migration 84
new features 83
ECU Interface Manager
migration 87
new features 89
ModelDesk
new features 91
MotionDesk
migration 95
new features 94
MATLAB
supported releases 183
MicroAutoBox
new features 102
migrating
ASM Diesel Engine Blockset 34
ASM Diesel Exhaust Blockset 35
ASM Drivetrain Basic Blockset 35
ASM Electric Components Blockset 36
ASM Engine Gasoline Basic Blockset 39
ASM Engine Gasoline Blockset 41
ASM Environment Blockset 37
ASM Optimizer Blockset 42
ASM Traffic Blockset 42
ASM Trailer Blockset 43
ASM Truck Blockset 45
ASM Turbocharger Blockset 46
migrating
ASM Vehicle Dynamics Blockset 47
ControlDesk 3.x 58
SystemDesk 127
AutomationDesk 30
ControlDesk Next Generation 75
dSPACE FlexRay Configuration Package 82
dSPACE Python Extensions 84
ECU Interface Manager 87
Model Compare 90
MotionDesk 95
Real-Time Testing 99
RTI 103
RTI Bypass Blockset 107
RTI CAN MultiMessage Blockset 110
RTI FPGA Programming Blockset 115
RTI LIN MultiMessage Blockset 112
Model Compare
migration 90
new features 89
ModelDesk
new features 91
MotionDesk
migration 95
new features 94
new features
ASM blocksets 32
ASM Diesel Engine Blockset 33
ASM Engine Gasoline Basic Blockset 38
ASM Engine Gasoline Blockset 40
ASM Environment Blockset 37
ASM Trailer Blockset 43
ASM Vehicle Dynamics Blockset 47
AutomationDesk 27
ControlDesk 58
ControlDesk Next Generation 60
dSPACE FlexRay Configuration Package 79
dSPACE HIL API .NET 77
dSPACE Offline Simulator 179
dSPACE Python Extensions 83
ECU Interface Manager 85
MicroAutoBox 102
Model Compare 89
ModelDesk 91
MotionDesk 94
Real-Time Testing 99
RTI Bypass Blockset 105
RTI CAN MultiMessage Blockset 109
RTI FPGA Programming Blockset 113
new features
  RTI LIN MultiMessage Blockset 111
  RTI/RTI-MP 101
  RTLib 101
  SystemDesk 118
not supported MATLAB features 102

P
  product overview 16

R
  Real-Time Testing
    migration 99
  requirements
    host PC software
    operating system 184
  RTI Bypass Blockset
    migration 107
  new features 105
  RTI CAN MultiMessage Blockset
    migration 110
  new features 109
  RTI FPGA Programming Blockset
    migration 115
  new features 113
  RTI LIN MultiMessage Blockset
    migration 112
  new features 111
  RTI/RTI-MP
    new features 101
  RTLib
    new features 101

S
  supported MATLAB releases 183
  system requirements
    operating system 184
  SystemDesk
    migrating 127
    new features 118

T
  TargetLink
    API commands
      changes 175
      AUTOSAR features, new
        compatibility mode 146
        component-based
          development 146
        exchanging SWCs as object
          code 146
        frame model generation,
          improvements 146
        frame models, updating 146
    AUTOSAR features, new
      hook functions, new 142
      supported releases 146
      code changes
        migration 168
        code efficiency, improvement of 144
        code generator options
          backward compatibility 174
          changed default value 174
      hook functions
        AUTOSAR 142
        new 142
        migrating to
          new version 158
      migration
        AUTOSAR related 176
        code changes 168
        obsolete limitations 177
        various aspects 176
      new API commands
        V-ECU generation 142
        virtual ECU generation 142
        new code generator options 138
      new features 132
      AUTOSAR 146
      code generation 132
      component-based
        development 133
      compute-through-overflow, optional
        deactivation 134
      custom code, improvements to 134
      general changes 145
      general enhancements 145
      modular development 133
      virtual ECU (V-ECU) generation
        new version
        migrating to 158
    target support
      discontinued compiler versions 136
      discontinued evaluation boards 136
      new compiler versions 136
      new evaluation boards 136
      supported targets 136
  TargetLink Data Dictionary
    API commands
      changes 175
      new commands 155
      discontinued features 160
      migrating to
        new version 158
      migration 160
      discontinued documentation 160
      upgrading existing data
        dictionaries 164, 165
      upgrading from version 2.x 167
    compare and merge 151
    data management 152
    DD IncludeFileGroup objects 155
diff and merge 151
  Loading one DD to different
    workspaces 155
  Object Explorer improvements 155
  points of inclusion 152
  Property Value List
    improvements 155
    new version
    migrating to 158

V
  version history 16

W
  Windows 64-Bit
    limitations 189
  Windows 7
    limitations 188
  Windows Vista
    limitations 188