dSPACE Release

New Features and Migration

Release 2013-B – November 2013
How to Contact dSPACE

To contact dSPACE if you have problems and questions, fill out the support request form provided on the website at http://www.dspace.com/go/supportrequest. The request form helps the support team handle your difficulties quickly and efficiently. In urgent cases contact dSPACE via phone:

- General Technical Support: +49 5251 1638-941
- TargetLink Support: +49 5251 1638-700

Software Updates and Patches

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

Important Notice

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

© Copyright 2000 - 2013 by:
dSPACE GmbH
Rathenaumstraße 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 11
   Document Symbols and Conventions ........................................... 12
   Accessing Online Help and PDF Files ......................................... 14

Overview of dSPACE Release 2013-B 15
   General Enhancements and Changes ............................................... 16
   64-Bit Version of RCP and HIL Software .................................... 19
   Product Version Overview .............................................................. 22
   New Product Key Features .............................................................. 26

Aspects When Migrating from Previous Releases 33
   Migrating to dSPACE Release 2013-B ............................................. 33

Migrating Python Scripts from Python 2.5 to Python 2.7 35
   Overview of the Main Changes ....................................................... 36
   Main Changes in Python 2.7 .............................................................. 36
   Main Changes of Python Handling in dSPACE Software ..................... 37
   General Information on Using Python Installations ......................... 41
   Required Modifications in Your Scripts .......................................... 42
   Migrating Changes in the Python Core Package ............................... 42
   Migrating Changes in the PyWin32 Package .................................... 45
   Using the dSPACE Library .............................................................. 46
   Enhancements to the Standard Python 2.7 Distribution ..................... 47

AutomationDesk 49
   New Features of AutomationDesk 4.0 ............................................. 49
   Migrating to AutomationDesk 4.0 ...................................................... 55

Automotive Simulation Models (ASM) 61
   All ASM Blocksets ............................................................... 62
   Migration of All ASM Blocksets .................................................... 62
   ASM Base InCylinder Blockset ....................................................... 63
New Features of ASM Base InCylinder Blockset 1.8 ............ 63
Migrating to ASM Base InCylinder Blockset 1.8 ............... 63
ASM Diesel Engine Blockset .................................................. 65
New Features of ASM Diesel Engine Blockset 1.6.3 .......... 65
Changes in the ASM Diesel Engine Demo Model ................ 65
Migrating to ASM Diesel Engine Blockset 1.6.3 ............... 66
ASM Diesel Exhaust Blockset ................................................... 67
ASM Diesel Exhaust Blockset 1.3 ......................................... 67
Changes in the ASM Diesel Exhaust Demo Model .......... 68
Migrating to ASM Diesel Exhaust Blockset 1.3 ............... 69
ASM Diesel InCylinder Blockset ............................................ 70
Changes in the ASM Diesel InCylinder Demo Model ...... 70
Migrating to ASM Diesel InCylinder Blockset 1.7 .......... 70
ASM Drivetrain Basic Blockset ............................................. 71
Migrating to ASM Drivetrain Basic Blockset 2.0.1 ........ 71
ASM Electric Components Blockset ..................................... 72
New Features of ASM Electric Components Blockset 2.5 .... 72
Changes in the ASM Electric Components Demo Model .... 73
ASM Environment Blockset ................................................... 74
New Features of ASM Environment Blockset 3.1 ............ 74
Migrating to ASM Environment Blockset 3.1 ............... 74
ASM Engine Gasoline Blockset ............................................ 75
New Features of ASM Engine Gasoline Blockset 2.4.3 .... 75
ASM Gasoline InCylinder Blockset ....................................... 76
New Features of ASM Gasoline InCylinder Blockset 1.7 .... 76
Migrating to ASM Gasoline InCylinder Blockset 1.7 ....... 76
ASM Traffic Blockset ............................................................ 77
Changes in the ASM Traffic Demo Model ....................... 77
ASM Trailer Blockset ............................................................ 78
New Features of ASM Trailer Blockset 2.2 ....................... 78
Changes in the ASM Trailer Demo Model .................... 79
Migrating to ASM Trailer Blockset 2.2 ....................... 79
ASM Truck Blockset ............................................................. 80
New Features of ASM Truck Blockset 2.1 ......................... 80
Changes in the ASM Truck Demo Model ...................... 81
Migrating to ASM Truck Blockset 2.1 ....................... 81
ASM Turbocharger Blockset .................................................. 82
New Features of ASM Turbocharger Blockset 2.1.2 ........... 82
Migrating to ASM Turbocharger Blockset 2.1.2 .............. 82
ASM Vehicle Dynamics Blockset ......................................... 84
New Features of ASM Vehicle Dynamics Blockset 2.5 .... 84
Changes in the ASM Vehicle Dynamics Demo Model .... 85
dSPACE FlexRay Configuration Package 129
New Features of dSPACE FlexRay Configuration Package 3.2 129
Migrating to dSPACE FlexRay Configuration Package 3.2 131

dSPACE Python Extensions 133
New Features of dSPACE Python Extensions 1.5 133
Migrating to dSPACE Python Extensions 1.5 134

ECU Interface Manager 137
New Features of ECU Interface Manager 1.4 137
Migrating to ECU Interface Manager 1.4 140

Firmware Manager 141
Features of Firmware Manager 1.0 141

Model Compare 143
New Features of Model Compare 2.4 143
Migration to Model Compare 2.4 144

ModelDesk 145
New Features of ModelDesk 3.1 145
Migration to ModelDesk 3.1 147

MotionDesk 149
New Features of MotionDesk 3.3 149
Migration to MotionDesk 3.3 150

Real-Time Testing 151
Migration to Real-Time Testing 2.2 151

RTI/RTI-MP and RTLib 153
New Features of RTI/RTI-MP and RTLib 153
Migration Aspects of RTI/RTI-MP and RTLib 156

RTI Bypass Blockset 163
New Features of the RTI Bypass Blockset 3.1 163
Migration to RTI Bypass Blockset 3.1 166

RTI CAN MultiMessage Blockset 167
New Features of the RTI CAN MultiMessage Blockset 2.9 167
Contents

Migrating to RTI CAN MultiMessage Blockset 2.9 .......................... 169

RTI LIN MultiMessage Blockset .............................. 171
   New Features of the RTI LIN MultiMessage Blockset 2.2 ............ 171
   Migrating to RTI LIN MultiMessage Blockset 2.2 ................. 172

RTI FPGA Programming Blockset .............................................. 173
   New Features of the RTI FPGA Programming Blockset 2.6 ......... 173
   Migrating to RTI FPGA Programming Blockset 2.6 ................... 176

RTI Watchdog Blockset .......................................................... 179
   Features of RTI Watchdog Blockset 1.0 ................................. 179
   Migrating to RTI Watchdog Blockset 1.0 ............................... 180

SystemDesk 4.x ........................................................................ 181
   New Features of SystemDesk 4.1 ........................................... 182
      New General Features ...................................................... 182
      Modeling Software Architectures ..................................... 182
      Modeling Systems .......................................................... 184
      Using Validation Rule Sets .............................................. 185
      Improved Data Exchange Configuration ............................ 186
   Migrating to SystemDesk 4.1 ................................................. 190
      Migrating to SystemDesk 4.1 ........................................... 190

TargetLink ............................................................................. 191
   New Features of TargetLink 3.5 and TargetLink Data Dictionary 3.5 192
      Modeling in Simulink or Stateflow .................................... 192
      Newly Supported Simulink Blocks .................................... 192
      Enumeration Support ...................................................... 193
      Improved System Preparation ......................................... 194
      Improved Stateflow Support ........................................... 195
      Code Generation Core Functionality ................................. 195
      MISRA-C:2004 compliance ............................................. 195
      Improved Code Efficiency .............................................. 196
      Data Dictionary and Datamanagement ............................... 197
      Enhanced Usability of the Data Dictionary Manager .............. 197
      Further Improvements to the Data Dictionary ..................... 201
      New DD MATLAB API Commands ................................. 201
      AUTOSAR ................................................................. 202
      Supported AUTOSAR Releases ..................................... 202
      Multiple Instantiation of Software Components ................. 203
New Features and Migration November 2013

Per Instance Calibratable Parameters .............................................................. 204
Support of Code Generation Units Containing Only Operation Calls .......................... 205
Easier Specification of AUTOSAR Objects in the Data Dictionary Manager .......................... 205
Improved Restart Function Behavior for Incrementally Generated Runnable Code .................... 206
Improved Container Management ................................................................. 207
Testing Support ............................................................................................. 208
Signal Injection/Tunneling ........................................................................... 208
Third-Party Tool Support for Code Coverage Measurements ........................................... 208
Enhanced Overflow Detection ........................................................................ 209
Changes in the Target Simulation Modules ......................................................... 210
Code Generator Options ................................................................................ 211
Code Generator Options ................................................................................ 211
Toolchain Integration ..................................................................................... 213
Handling Customization Files .......................................................................... 213
Enhanced Handling and Restoring of Demo Models ............................................... 214
TargetLink Simulation Module Extension Packages .................................................. 215
Documentation ............................................................................................. 216
Improved Documentation ................................................................................ 216
Other ............................................................................................................. 217
New Code Generation Report ........................................................................... 217
General Enhancements and Changes ....................................................................... 219
API Commands ............................................................................................. 220
New API Commands ....................................................................................... 220
Hook Functions ............................................................................................. 221
New Hook Functions ....................................................................................... 221
Migrating to TargetLink 3.5 and TargetLink Data Dictionary 3.5 ................................. 222
Data Dictionary and Datamanagement ....................................................................... 224
Migrating to TargetLink Data Dictionary 3.5 ......................................................... 225
How to Upgrade a Data Dictionary Without Included DD Files ................................. 228
How to Upgrade a Data Dictionary With Included DD Files ...................................... 229
How to Upgrade Libraries and Models from TargetLink 2.x to 3.x ............................... 231
Code Changes ................................................................................................ 232
Code Changes ................................................................................................ 232
AUTOSAR ........................................................................................................ 236
AUTOSAR-Related Migration Aspects ................................................................... 236
Documentation ............................................................................................... 238
Documentation Changes .................................................................................... 238
Contents

Other........................................................................................239
Various Migration Aspects.......................................................239
API Commands.........................................................................240
  Changes in TargetLink and TargetLink Data Dictionary API
    Functions ............................................................................240
Hook Functions.........................................................................240
  Hook Function Templates.....................................................240
Obsolete...................................................................................241
  Discontinued Data Dictionary Features ...............................241
  Obsolete Limitations...........................................................242
Changes in Future TargetLink Versions.......................................243
  To be Discontinued...............................................................243
    Changes...............................................................................243
VEOS........................................................................................245
  New Features of VEOS 3.1 ....................................................245
    Migrating to VEOS 3.1........................................................247
Compatibility Information.........................................................249
  Supported MATLAB Releases............................................250
  Operating System..............................................................251
  Run-Time Compatibility of dSPACE Software.........................253
  Limitations for 64-Bit Windows Operating Systems in
    Combination with 32-Bit dSPACE Software.........................254
  Limitations for Products on 64-bit dSPACE DVD....................254
  Limitations for Windows 7 ...................................................257
Index.......................................................................................259
About This Document

Contents

This document informs you about the new features of all the dSPACE software products in Release 2013-B. It also gives you an overview of software products with no or minor changes. There are instructions on migrating from earlier dSPACE releases, especially from earlier product versions, if required.

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>Document Symbols and Conventions</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessing Online Help and PDF Files</td>
<td>14</td>
</tr>
</tbody>
</table>
Document Symbols and Conventions

Symbols

The following symbols may be used in this document.

- !: Indicates a general hazard that may cause personal injury of any kind if you do not avoid it by following the instructions given.
- ⚡: Indicates the danger of electric shock which may cause death or serious injury if you do not avoid it by following the instructions given.
- ⚠️: Indicates a hazard that may cause material damage if you do not avoid it by following the instructions given.
- 🚨: Indicates important information that should be kept in mind, for example, to avoid malfunctions.
-💡: Indicates tips containing useful information to make your work easier.

Naming conventions

The following abbreviations and formats are used in this document:

- %name%: Names enclosed in percent signs refer to environment variables for file and path names.
- < >: Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.
- ⚪️: Precedes the document title in a link that refers to another document.
- 📚: Indicates that a link refers to another document, which is available in dSPACE HelpDesk.

Special folders

Some software products, for example, ControlDesk Next Generation and AutomationDesk, use the following special folders:

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

The location of the Common Program Data folder depends on the operating system.

- **Windows XP**:
  ```
  \Documents and Settings\All Users\Application Data\dSPACE\<InstallationGUID>\<ProductName>
  ```
- Windows 7:
  .\ProgramData\dSPACE\<InstallationGUID>\<ProductName>

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

The location of the *Documents folder* depends on the operating system.

- Windows XP:
  .\Documents and Settings\<User>\My Documents\dSPACE\<ProductName>\<VersionNumber>

- Windows 7:
  .\Users\<User>\My Documents\dSPACE\<ProductName>\<VersionNumber>

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

The location of the *Local Program Data folder* depends on the operating system.

- Windows XP:
  .\Documents and Settings\<User>\Local Settings\Application Data\dSPACE\<InstallationGUID>\<ProductName>

- Windows 7:
  .\Users\<User>\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>
## Accessing Online Help and PDF Files

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

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

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

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

![Context-sensitive help is not available in all software products.](image)

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

### PDF files
You can access the PDF files as follows:

**dSPACE HelpDesk**
Click the PDF link at the beginning of a document:
Overview of dSPACE Release 2013-B

Objective
Gives you an overview of the new key features in Release 2013-B and information about unchanged products.

Where to go from here
Information in this section

| General Enhancements and Changes                  | 16 |
| 64-Bit Version of RCP and HIL Software            | 19 |
| Product Version Overview                          | 22 |
| New Product Key Features                          | 26 |
# General Enhancements and Changes

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

**New release name**
As of dSPACE Release 2013-A, releases have a new name format. The release name now consists of the release date and a letter, for example, 2013-B to specify the second dSPACE Release in 2013.

**New Python support**
dSPACE software products which are using Python, for example, for their automation interfaces, now support Python 2.7. For further information and required migration steps, refer to Migrating Python Scripts from Python 2.5 to Python 2.7 on page 35.

**RCP and HIL software support for MATLAB (64-bit)**
The following RCP and HIL software products are now also available as product versions that support MATLAB® 64-bit versions:
- RTI Ethernet UDP Blockset
- RTI XCP on Ethernet Blockset
- RTI RapidPro Control Unit Blockset

For detailed information on which products are available, refer to 64-Bit Version of RCP and HIL Software on page 19.

**Distribution of 32-bit and 64-bit software**
As of dSPACE Release 2013-A, the dSPACE software is distributed on two DVDs, each containing the same content but with the following differences:
- One 32-bit DVD containing only 32-bit dSPACE software products, for example, to support 32-bit MATLAB versions
- One 64-bit DVD containing:
  - All MATLAB-related dSPACE products which have been ported to support 64-bit MATLAB versions
  - 32-bit versions of MATLAB-related dSPACE products which were not yet ported to support 64-bit MATLAB versions

Exception for MATLAB-related products from the RCP and HIL software package: The 64-bit DVD does not contain 32-bit versions of unported MATLAB-related products because these products are not functional in combination with 64-bit MATLAB versions. If you want to use such a product from the RCP and HIL software package, you have to install and use the RCP and HIL software package from the 32-bit DVD.
All 32-bit dSPACE products that do not relate to MATLAB (for example, ControlDesk Next Generation).

You can therefore install dSPACE software from one DVD without changing DVDs during the installation procedure.

For a complete list of all dSPACE products contained on dSPACE Release 2013-B as 64-bit versions, refer to Limitations for Products on 64-bit dSPACE DVD on page 254.

Restrictions when working with dSPACE HelpDesk

dSPACE HelpDesk is installed in release-specific folders in C:\Program Files\Common Files\dSPACE on a 32-bit operating system and in C:\Program Files(x86)\Common Files\dSPACE on a 64-bit operating system. For example, if you have installed products from dSPACE Release 2013-B and products from dSPACE Release 2013-A, there are two dSPACE HelpDesks available.

With dSPACE Release 2013-B RCP and HIL software also accesses dSPACE HelpDesk in the above-mentioned folders.

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.

After you install dSPACE Release 2013-B, you can find the user documentation for the following products in dSPACE HelpDesk 2013-A or earlier:

- SystemDesk 3.2

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

To facilitate finding user documentation in the `print` folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the `online` folder.

For further information, refer to the product-specific migration descriptions.

## Release update

The printed user documentation is not delivered with Release 2013-B 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.

## Software support discontinuation

The following products are provided up to dSPACE Release 2013-A. As of dSPACE Release 2013-B, they have been discontinued:

- ControlDesk 3.x
  
  ControlDesk Next Generation is the successor of ControlDesk 3.x. For information on ControlDesk Next Generation and instructions on how to migrate ControlDesk 3.x experiments, refer to the ControlDesk Next Generation’s user documentation.

- CLIB

- Test Automation Blockset

- The following modules from dSPACE Test Automation Python Modules:
  
  - rtplib
  - sdmlib
  - matlablib
  - rs232lib

For detailed migration instructions, refer to *Migrating Python Scripts from Python 2.5 to Python 2.7* on page 35.
64-Bit Version of RCP and HIL Software

Objective

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

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

In general, the RCP and HIL (64-bit) software contains the same products as the RCP and HIL software available on the dSPACE Release 2013-B (32-bit) DVD. However, not all MATLAB-related RCP and HIL software products have been ported to MATLAB x64 so far. These products are therefore not all contained in RCP and HIL (64-bit).

<table>
<thead>
<tr>
<th>RCP and HIL Product</th>
<th>Contained on 64-bit DVD as ...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64-bit version</td>
</tr>
<tr>
<td>RTI and RTI-MP</td>
<td>✓</td>
</tr>
<tr>
<td>RTI Gigalink Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTI CAN Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTI CAN MultiMessage Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTI LIN MultiMessage Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTI FlexRay Configuration Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTI FPGA Programming Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTI AUTOSAR Package</td>
<td>_ 1)</td>
</tr>
<tr>
<td>RTI Ethernet UDP Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTI XCP on Ethernet Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTI RapidPro Control Unit Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTI Bypass Blockset</td>
<td>_ 1)</td>
</tr>
<tr>
<td>RTI Watchdog Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>MLIB/MTRACE</td>
<td>_ 2)</td>
</tr>
<tr>
<td>ConfigurationDesk - Configuration Version</td>
<td>-</td>
</tr>
<tr>
<td>ConfigurationDesk - Implementation Version</td>
<td>-</td>
</tr>
<tr>
<td>FlexRay Configuration Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>FlexRay Configuration Tool</td>
<td>-</td>
</tr>
<tr>
<td>ModelDesk</td>
<td>-</td>
</tr>
<tr>
<td>Automotive Simulation Models</td>
<td>✓</td>
</tr>
<tr>
<td>MotionDesk</td>
<td>-</td>
</tr>
<tr>
<td>MotionDesk Blockset</td>
<td>✓</td>
</tr>
</tbody>
</table>
Both RCP and HIL software versions can be installed on your computer, but only one can be active at the same time. Use the dSPACE Installation Manager to switch between them.

For information on the 64-bit support of all dSPACE products and further limitations, refer to Limitations for Products on 64-bit dSPACE DVD on page 254.

<table>
<thead>
<tr>
<th>RCP and HIL Product</th>
<th>Contained on 64-bit DVD as ...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64-bit version</td>
</tr>
<tr>
<td>Tools: Flight Rec Data Merger</td>
<td>–</td>
</tr>
<tr>
<td>Further products of RCP and HIL software package</td>
<td>–</td>
</tr>
</tbody>
</table>

1) Support of 64-bit MATLAB versions is planned for future releases. Use the 32-bit DVD (with 32-bit MATLAB version support) to install this product.

2) Support of 64-bit MATLAB versions is not planned for future releases. Use the 32-bit DVD (with 32-bit MATLAB version support) to install this product. Note, that MLIB/MTRACE will be discontinued with the next dSPACE release.

3) Supports 64-bit MATLAB versions.

Supported MATLAB versions

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

- MATLAB R2012b (64-bit)
- MATLAB R2013a (64-bit)
- MATLAB R2013b (64-bit)

See also Supported MATLAB Releases on page 250.

Supported MEX compiler

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

This compiler is a free download from Microsoft. The compiler requires the .NET framework 4.0, which is also available at no charge from Microsoft. Download links and instructions for the compiler and framework can be found at http://www.mathworks.com/support/compilers/R2012b/win64.html.

You need to install this compiler and configure it in MATLAB as a MEX compiler if you intend to use RCP and HIL products that require a MEX compiler: e.g., RTI CAN MultiMessage Blockset, RTI LIN MultiMessage Blockset, or Automotive Simulation Models.
System requirements

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

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

See also Operating System on page 251.
## 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>AutomationDesk</td>
<td>7.3: 3.5, 7.4: 3.6, 2013-A: 3.6p2, 2013-B: 4.0</td>
</tr>
<tr>
<td>Automotive Simulation Models</td>
<td>2013-B: 5.1</td>
</tr>
<tr>
<td>Container Manager</td>
<td>7.3: 3.1, 7.4: 3.2, 2013-A: 3.2, 2013-B: 4.1</td>
</tr>
<tr>
<td>ControlDesk 3.x&lt;sup&gt;(1)&lt;/sup&gt;</td>
<td>7.3: 3.7.3, 7.4: 3.7.4, 2013-A: 3.7.5, 2013-B: - Discontinued</td>
</tr>
<tr>
<td>ControlDesk Next Generation&lt;sup&gt;(2)&lt;/sup&gt;</td>
<td>7.3: 4.2.1, 7.4: 4.3, 2013-A: 5.0, 2013-B: 5.1</td>
</tr>
<tr>
<td>DCI Configuration Tool</td>
<td>7.3: 2.5.1, 2013-A: 2.5.1&lt;sup&gt;(3)&lt;/sup&gt;, 2013-B: 3.0</td>
</tr>
<tr>
<td>dSPACE CAN API</td>
<td>7.3: 2.5.1, 7.4: 2.6, 2013-A: 2.6, 2013-B: 2.6</td>
</tr>
<tr>
<td>dSPACE ECU Flash Programming Tool</td>
<td>7.3: 2.2.1, 2013-A: 2.2.1, 2013-B: 2.2.3</td>
</tr>
</tbody>
</table>
## Overview of dSPACE Release 2013-B

### New Features and Migration November 2013

<table>
<thead>
<tr>
<th>Product</th>
<th>dSPACE Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.3</td>
</tr>
<tr>
<td>dSPACE HIL API .NET</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>dSPACE Python Extensions</td>
<td>1.2</td>
</tr>
<tr>
<td>ECU Interface Manager</td>
<td>1.1</td>
</tr>
<tr>
<td>Firmware Manager</td>
<td></td>
</tr>
<tr>
<td>Model Compare</td>
<td>2.2</td>
</tr>
<tr>
<td>ModelDesk</td>
<td>2.5</td>
</tr>
<tr>
<td>MotionDesk</td>
<td>3.0</td>
</tr>
<tr>
<td>MotionDesk Blockset</td>
<td>2.2</td>
</tr>
<tr>
<td>Real-Time Testing</td>
<td>1.9</td>
</tr>
<tr>
<td>RTI</td>
<td>6.9</td>
</tr>
<tr>
<td>RTI-MP</td>
<td>6.5</td>
</tr>
<tr>
<td>RTI AUTOSAR Package</td>
<td>1.3.1</td>
</tr>
<tr>
<td>RTI Bypass Blockset</td>
<td>2.9</td>
</tr>
</tbody>
</table>

- See `dSPACE HIL API .NET` on page 127.
- See `dSPACE Python Extensions` on page 133.
- See `ECU Interface Manager` on page 137.
- See `Firmware Manager` on page 141.
- See `Model Compare` on page 143.
- See `ModelDesk` on page 145.
- See `MotionDesk` on page 149.
- See `Real-Time Testing` on page 151.
- See `RTI/RTI-MP and RTLib` on page 153.
<table>
<thead>
<tr>
<th>Product</th>
<th>dSPACE Release</th>
<th>7.3</th>
<th>7.4</th>
<th>2013-A</th>
<th>2013-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTI CAN Blockset</td>
<td></td>
<td>2.8</td>
<td>2.8.1</td>
<td>3.0</td>
<td>3.1</td>
</tr>
<tr>
<td>RTI CAN MultiMessage Blockset</td>
<td></td>
<td>2.6</td>
<td>2.7</td>
<td>2.8</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See RTI CAN MultiMessage Blockset on page 167.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI LIN MultiMessage Blockset</td>
<td></td>
<td>1.9</td>
<td>2.0</td>
<td>2.1</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See RTI LIN MultiMessage Blockset on page 171.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI RapidPro Control Unit Blockset</td>
<td></td>
<td>1.12</td>
<td>2.1</td>
<td>2.1.1</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See RTI/RTI-MP and RTLib on page 153.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI FPGA Programming Blockset</td>
<td></td>
<td>2.3</td>
<td>2.4</td>
<td>2.5</td>
<td>2.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See RTI FPGA Programming Blockset on page 173.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI Ethernet (UDP) Blockset</td>
<td></td>
<td>1.2</td>
<td>1.2</td>
<td>1.2.1</td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See RTI/RTI-MP and RTLib on page 153.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI Watchdog Blockset</td>
<td></td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See RTI Watchdog Blockset on page 179.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNECT server</td>
<td></td>
<td>--</td>
<td>--</td>
<td>1.1</td>
<td>1.2</td>
</tr>
<tr>
<td>SystemDesk 3.x&lt;sup&gt;5&lt;/sup&gt;</td>
<td></td>
<td>3.1p3</td>
<td>3.2</td>
<td>3.2</td>
<td>3.2&lt;sup&gt;5&lt;/sup&gt;)</td>
</tr>
<tr>
<td>SystemDesk 4.x&lt;sup&gt;7&lt;/sup&gt;</td>
<td></td>
<td>--</td>
<td>4.0</td>
<td>4.0</td>
<td>4.1&lt;sup&gt;8&lt;/sup&gt;)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See SystemDesk 4.x on page 181.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TargetLink/TargetLink Data Dictionary</td>
<td></td>
<td>3.3</td>
<td>3.4</td>
<td>3.4</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>See TargetLink on page 191.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If you have not updated regularly, refer to the New Features and Migration documents for the dSPACE Releases listed above for information about the new features and necessary migration steps.
New Product Key Features

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

Information in this topic
AutomationDesk on page 26
Automotive Simulation Models (ASMs) on page 27
ConfigurationDesk on page 27
Container management on page 27
ControlDesk Next Generation on page 27
DCI Configuration Tool on page 28
dSPACE FlexRay Configuration Package on page 28
ECU Interface Manager on page 29
Model Compare on page 29
ModelDesk on page 29
MotionDesk on page 29
Python Extensions on page 29
RTI, RTI-MP and RTLib on page 30
RTI Bypass Blockset on page 30
RTI CAN MultiMessage Blockset on page 30
RTI LIN MultiMessage Blockset on page 30
RapidPro Control Unit RTI Blockset on page 31
RTI Ethernet (UDP) Blockset on page 31
RTI FPGA Programming Blockset on page 31
RTI Watchdog Blockset on page 31
SystemDesk 4.x on page 31
TargetLink on page 31
VEOS on page 32

AutomationDesk
The new key features of AutomationDesk are:

- New libraries for platform management and remote-control of ModelDesk
- Enhancements to the existing libraries, e.g., the GetVerdicts and Invoke blocks in the Main Library
- Dialogs library now supports the offline operation mode
- Ribbon menus for the user interface
- Consistent verdict handling in all libraries
- New pane for setting library favorites
- Enhanced Python Editor and Value Editor features

For details on the new features, refer to New Features of AutomationDesk 4.0 on page 49.
### Automotive Simulation Models (ASMs)
For details on the new features, refer to *Automotive Simulation Models (ASM)* on page 61.

### ConfigurationDesk
For details on the new features, refer to *ConfigurationDesk – Implementation* on page 87.

### Container management
The new key feature of container management is:
- Improved exchange of software components between SystemDesk 4.1 and TargetLink

For details on the new features, refer to *New Features of Container Management* on page 95.

### ControlDesk Next Generation
The new key features of ControlDesk Next Generation (ControlDesk 5.1) are:
- **Platform management and platforms/devices:**
  - Option to rename platforms/devices
  - Support of Vector Informatik’s VN8900 system (CAN, LIN and FlexRay)
  - Adjustable period for variable observation
  - Video Capturing device: Replay of recorded videos
- **Visualization and instruments:**
  - Importing and exporting variable connections
  - User-defined tooltips for instruments
  - Adjusting the z order of instruments
  - Custom default background color for layouts
  - Steering Controller: Force feedback support
  - Variable Array: Alternating row colors
  - Static Text: User-defined text angle
- **Measurement and recording:**
  - Simulation time groups to improve synchronization accuracy
  - Usability improvements to the Measurement Configuration controlbar
  - Import and export of recorders
- **ECU Diagnostics:** Support of the ASAM MCD-2 D v2.2.0 ODX standard
- **Signal Editor:** Exporting signal descriptions and single signals to the CSV file format
Failure simulation: Extended support for switching multiple failures on SCALEXIO

Automation:
- Support of Python 2.7
- Automatic execution of Python extension scripts when ControlDesk Next Generation starts up
- Migration from ControlDesk 3.x: Migration of ControlDesk 3.x automation is documented

For details on the new features, refer to New Features of ControlDesk Next Generation (ControlDesk 5.1) on page 98.

**DCI Configuration Tool**
The new key features of the DCI Configuration Tool are:
- Checks to perform start-up diagnostics for DCI-GSI2s
- Usability improvements

For details on the new features, refer to New Features of the DCI Configuration Tool 3.1 on page 123.

**dSPACE FlexRay Configuration Package**
The new key features of the dSPACE FlexRay Configuration Tool are:
- Support of AUTOSAR System Template 4.1.1
- Improved handling in General Properties dialog

The new key feature of the RTI FlexRay Configuration Blockset is:
- Switchable check for RTI FlexRay Configuration blocks from custom libraries during update process

The new key feature of the FlexRay Configuration Blockset is:
- Switchable check for FlexRay Configuration blocks from custom libraries during update process

The new key feature of the FlexRay Replay Script Generator is:
- Support of Python 2.7

For details on the new features, refer to New Features of dSPACE FlexRay Configuration Package 3.2 on page 129.
### ECU Interface Manager
The new key features of the ECU Interface Manager are:
- Support of Freescale MPC57xx microcontrollers
- Display of the remaining memory resources
- Support for grouping functions in the Library pane
- Macros for naming A2L/binary files
- Support of the Intel HEX file format

For details on the ECU Interface Manager, refer to *New Features of ECU Interface Manager 1.4* on page 137.

### Model Compare
The new key features of Model Compare are:
- Easy access to comparison settings via favorites
- Traceability from model to Model Compare
- Enhanced report generation options

For details on the new features, refer to *New Features of Model Compare 2.4* on page 143.

### ModelDesk
The new key features of ModelDesk are:
- Synchronized platform management
- Road Generator provides configuration data for a multi-PC solution in MotionDesk

For details on the new features, refer to *New Features of ModelDesk 3.1* on page 145.

### MotionDesk
The new key features of MotionDesk are:
- Scene generation in a multi-PC solution
- Improvements for the visualization

For details on the new features, refer to *New Features of MotionDesk 3.3* on page 149.

### Python Extensions
The Python modules now support Python 2.7.

For details on the new features, refer to *dSPACE Python Extensions* on page 133.
<table>
<thead>
<tr>
<th>Module</th>
<th>New Key Features</th>
</tr>
</thead>
</table>
| **RTI, RTI-MP and RTLib** | The new key features of RTI, RTI-MP and RTLib are:  
  - New MP modeling mode  
  - Software support of the new DS1513 I/O Board for MicroAutoBox II  
  For details on the new features, refer to *New Features of RTI/RTI-MP and RTLib* on page 153. |
| **RTI Bypass Blockset** | The new key features of the RTI Bypass Blockset are:  
  - Support of further target processor families for internal bypassing  
  - Memory fragmentation for internal bypassing  
  - Support of VLE code for Power-PC (MPC5xxx) microcontroller architectures  
  - Influencing and extending generated A2L file elements by means of a MATLAB configuration structure  
  - Support of Intel HEX file format  
  The new key feature of the RTI Bypass Blockset MATLAB API is:  
  - Support of enhancements to RTI Bypass Blockset  
  For details on the new features, refer to *New Features of the RTI Bypass Blockset 3.1* on page 163. |
| **RTI CAN MultiMessage Blockset** | The new key features of the RTI CAN MultiMessage Blockset are:  
  - Support of MicroAutoBox II with the new DS1513 I/O board  
  - Support of AUTOSAR System Template 4.1.1  
  - Addressing J1939 messages by their network addresses  
  - Manipulating gateway directions individually during run time  
  For details on the new features, refer to *New Features of the RTI CAN MultiMessage Blockset 2.9* on page 167. |
| **RTI LIN MultiMessage Blockset** | The new key features of the RTI LIN MultiMessage Blockset are:  
  - Support of MicroAutoBox II with the new DS1513 I/O board  
  - Support of AUTOSAR System Template 4.1.1  
  - Simulation of up to 63 LIN slave nodes in an RTILINMM configuration  
  For details on the new features, refer to *New Features of the RTI LIN MultiMessage Blockset 2.2* on page 171. |
### Overview of dSPACE Release 2013-B

**New Features and Migration November 2013**

<table>
<thead>
<tr>
<th>Blockset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RapidPro Control Unit RTI Blockset</strong></td>
<td>The blockset now supports 64-bit MATLAB versions.</td>
</tr>
<tr>
<td><strong>RTI Ethernet (UDP) Blockset</strong></td>
<td>The blockset now supports 64-bit MATLAB versions.</td>
</tr>
<tr>
<td><strong>RTI FPGA Programming Blockset</strong></td>
<td>The new key features of the RTI FPGA Programming Blockset are:</td>
</tr>
<tr>
<td></td>
<td>- Extended Xilinx® software support</td>
</tr>
<tr>
<td></td>
<td>- General enhancements, for example, group number display and variable buffer sizes</td>
</tr>
<tr>
<td></td>
<td>- Support of inter-FPGA communication with the DS5203 FPGA Board framework</td>
</tr>
<tr>
<td></td>
<td>- New ports for APU Slave (DS2655 FPGA Base Module) and ADC (DS2655M1 I/O Module)</td>
</tr>
<tr>
<td></td>
<td>For details on the new features, refer to New Features of the RTI FPGA Programmer Blockset 2.6 on page 173.</td>
</tr>
<tr>
<td><strong>RTI Watchdog Blockset</strong></td>
<td>For details on the new blockset, refer to Features of RTI Watchdog Blockset 1.0 on page 179.</td>
</tr>
<tr>
<td><strong>SystemDesk 4.x</strong></td>
<td>The key features of SystemDesk 4.1 are:</td>
</tr>
<tr>
<td></td>
<td>- Support of AUTOSAR 4.1.1, 4.0.3, and 4.0.2.</td>
</tr>
<tr>
<td></td>
<td>- Improved support for connecting software components</td>
</tr>
<tr>
<td></td>
<td>- Modeling systems with the System Manager</td>
</tr>
<tr>
<td></td>
<td>- Using validation rule sets</td>
</tr>
<tr>
<td></td>
<td>- Improved configuration of data exchange</td>
</tr>
<tr>
<td></td>
<td>For details on the features, refer to New General Features on page 182.</td>
</tr>
<tr>
<td><strong>TargetLink</strong></td>
<td>The new key features of TargetLink are:</td>
</tr>
<tr>
<td></td>
<td>- Support of Simulink enumeration data types</td>
</tr>
<tr>
<td></td>
<td>- Support of AUTOSAR 4.1.1</td>
</tr>
<tr>
<td></td>
<td>- Multiple instantiation of AUTOSAR SWCs</td>
</tr>
<tr>
<td></td>
<td>- Improved tool chain integration</td>
</tr>
<tr>
<td></td>
<td>For details on all the new features, refer to New Features of TargetLink 3.5 and TargetLink Data Dictionary 3.5 on page 192.</td>
</tr>
</tbody>
</table>
For details on the TargetLink migration aspects (TargetLink, TargetLink AUTOSAR module, TargetLink Data Dictionary), refer to Migrating to TargetLink 3.5 and TargetLink Data Dictionary 3.5 on page 222.

VEOS

The new key features of VEOS are:

- VEOS now supports 64-bit MATLAB versions.
- dSPACE Target for Offline Simulation: Real-Time Testing support always enabled
- VEOS Player: Improved user interface

For details on the new features, refer to VEOS on page 245.
Aspects When Migrating from Previous Releases

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

Migrating to dSPACE Release 2013-B

Objective
After you install Release 2013-B, some additional steps may be necessary.

Migrating from dSPACE Release 2013-A
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 7.4 or earlier
To migrate from dSPACE Release 7.4 or earlier to Release 2013-B, you also have to perform the migration steps of the intervening dSPACE Releases. All of the required migration steps can be done with Release 2013-B installed.

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

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

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

- In the installation folder of the current dSPACE HelpDesk, see C:\Program Files(x86)\Common Files\dSPACE\HelpDesk 2013-B\Print\PreviousReleases.
- On the dSPACE DVD, see \Doc\Print\PreviousReleases.
- Download them from http://www.dspace.com/go/migration. Here you can also find New Features and Migration documents for very early releases.
Migrating Python Scripts from Python 2.5 to Python 2.7

Objective

To migrate Python scripts from Python 2.5 to Python 2.7, you have to note the following points.

If you want to migrate from an earlier version of Python to Python 2.7, you also have to note the migration steps described, for example, in the AutomationDesk Guide or the ControlDesk Next Generation MCD-3 Automation Guide from earlier dSPACE releases.

Where to go from here

Information in this section

| Overview of the Main Changes | 36 |
| Required Modifications in Your Scripts | 42 |
Overview of the Main Changes

**Objective**
Gives you information on the main changes in Python 2.7 and the new dSPACE handling of Python.

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

<table>
<thead>
<tr>
<th>Main Changes in Python 2.7</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Changes of Python Handling in dSPACE Software</td>
<td>37</td>
</tr>
<tr>
<td>General Information on Using Python Installations</td>
<td>41</td>
</tr>
</tbody>
</table>

Main Changes in Python 2.7

**Objective**
Provides information on Python 2.7 and the reasons why dSPACE decided to use it.

**What's New documentation from the Python Software Foundation**
The following What's New documents for Python versions from Python 2.5 to Python 2.7 are available from the Python Software Foundation:
- What's New from Python 2.5 to 2.6
  http://docs.python.org/2.6/whatsnew/2.6.html
- What's New from Python 2.6 to 2.7
  http://docs.python.org/2.7/whatsnew/2.7.html
You can refer to the Python documentation provided by the Python Software Foundation for further information on migrating Python scripts from Python 2.5 to Python 2.7.

**Main reasons for using Python 2.7 instead of 2.5**
The main reasons for migrating to Python 2.7 are:
- Support of new operating systems
- Support for newer versions of extension modules.
- Picking up bug fixes
Why not migrate to Python 3.x?

dSPACE decided to use Python 2.7, because of the announced long-term support for running production systems based on Python 2.x. Python 3.x comes with a lot of incompatible changes that would greatly increase the amount of work needed for migration. Besides not all extension modules required by dSPACE software are available for Python 3.x.

Main Changes of Python Handling in dSPACE Software

Objective

Migrating to Python 2.7 also changed the Python handling in dSPACE software.

Changed installation of Python 2.7

Up to dSPACE Release 2013-A, the Python 2.5 installation distributed on the dSPACE DVD was a dSPACE setup with specific enhancements required for working with dSPACE software. It was installed to C:\Program Files (x86)\Common Files\dSPACE\Python25.

With Python 2.7, the Python installation on the dSPACE DVD is using the setups for the Python core and extension modules that are provided by the Python community. dSPACE-specific components are now installed as add-ons. This allows you more flexibility when using Python, for example, you can add and update Python packages without affecting the dSPACE installation. However, the dSPACE software will issue a warning if you use a package or package version that is not tested by dSPACE. If you install the Python 2.7 installation from the dSPACE DVD, it is installed on C:\Program Files (x86)\Python27 by default. You can also use Python 2.7 installed in another folder. The dSPACE installation detects an already installed Python 2.7 and checks whether the Python core and required extensions must be installed or updated.
You have to check whether the methods in your scripts to get the Python installation path must be adapted. For example, you can read the installation path from the Windows registry.

- Using 32-bit Windows operating system:
  HKLM\SOFTWARE\Python\PythonCore\2.7\InstallPath

- Using 64-bit Windows operating system:
  HKLM\SOFTWARE\Python\Wow6432Node\PythonCore\2.7\InstallPath

The Python 2.5 installation had its own online help in dSPACE HelpDesk. With Python 2.7, there is no Python user documentation in dSPACE HelpDesk. You have to open the standard user documentation, for example, via Start – All Programs – Python 2.7.

You can modify the Python installation more freely than before, but you have to guarantee that the dSPACE software works. It is therefore recommended not to remove Python packages from the Python distribution installed by dSPACE.

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

<table>
<thead>
<tr>
<th>Python Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python Core</td>
<td>2.7.5</td>
</tr>
<tr>
<td>PyWin32</td>
<td>218.10</td>
</tr>
<tr>
<td>Numpy</td>
<td>1.7.1</td>
</tr>
<tr>
<td>Matplotlib</td>
<td>1.2.1</td>
</tr>
<tr>
<td>WxPython</td>
<td>2.9.4.0</td>
</tr>
<tr>
<td>Py2exe</td>
<td>0.6.9</td>
</tr>
<tr>
<td>Comtypes</td>
<td>0.6.2</td>
</tr>
<tr>
<td>PIL</td>
<td>1.1.7</td>
</tr>
</tbody>
</table>
Migrating Python Scripts from Python 2.5 to Python 2.7

Usage of Python interpreter changed

Some dSPACE products previously used an embedded Python interpreter. With dSPACE Release 2013-B, these products now use the Python interpreter provided by the Python 2.7 installation. This interpreter offers the same functionality as the embedded interpreter, but some API calls behave differently. See the migration instructions for details.

The following dSPACE products use the Python interpreter:
- ControlDesk Next Generation
- ConfigurationDesk
- ModelDesk
- SYNECT

With this interpreter you can access the Python implementation of the following dSPACE products, if installed:
- dSPACE Python Extensions (for example, the Python modules of the Platform API Package)
- Real-Time Testing
- dSPACE CAN API Deliverables

Due to the fact that the Python interpreter is separated from the dSPACE products, it closes when you shut down Windows even if the dSPACE product is still running. You have to restart the Python interpreter if the dSPACE product requires access to it.

AutomationDesk uses an embedded interpreter. With this interpreter, you can access only dSPACE Python modules that are included in the AutomationDesk installation, such as rtpilib2 and matlablib2.

Discontinued dSPACE Python modules

When migrating your Python scripts to Python 2.7 and dSPACE Release 2013-B, note that some dSPACE Python modules have been discontinued.

The following dSPACE Python modules have been discontinued in dSPACE Release 2013-B:
- rtpilib
- matlablib
- rs232lib
- failuresimlib

New Features and Migration November 2013
The following new Python modules are available for some dSPACE releases. These are compatible with the old ones but provide additional features.

- `sdmlib`
- `officelibs` (word, excel, powerpoint)

For further information, refer to *Test Automation Python Modules Reference*.

The `sdmlib` Python module must be replaced either by the stimulus feature of the HIL API MAPort implementation or by Real-Time Testing scripts. Stimulus files in the new STZ format can be created by using the Signal Editor of ControlDesk Next Generation or by converting existing files via the `sdm2stz` converter. For download and instructions, refer to [http://www.dspace.com/go/sdm2stz](http://www.dspace.com/go/sdm2stz).

The `cdautomationlib` Python module must be replaced by other products provided by dSPACE, for example, the automation interface of ControlDesk Next Generation or the dSPACE HIL API Python implementation, refer to *ControlDesk Next Generation API Reference* and *dSPACE HIL API Python Implementation Document*. With the ControlDesk Next Generation API you can also replace the `failuresimlib` Python module.

The Python modules `wordlib` and `excellib` are provided as support downloads for compatibility reasons at [http://www.dspace.com/go/dSPACEOfficeLibs](http://www.dspace.com/go/dSPACEOfficeLibs).

---

**Discontinuation of ControlDesk 3.x**

In dSPACE Release 2013-B, ControlDesk 3.x has also been discontinued. Scripts for automating ControlDesk 3.x can be migrated to be used with ControlDesk Next Generation. For the required migration steps, refer to *ControlDesk Next Generation Migration of ControlDesk 3.x Automation*. 

---
## General Information on Using Python Installations

<table>
<thead>
<tr>
<th>Objective</th>
<th>The following information is relevant if you want to use both Python versions on your computer.</th>
</tr>
</thead>
</table>
| Using Python 2.5 and Python 2.7 in parallel | Both Python versions can be used in parallel on your computer with the following restrictions:  
- The file associations for PY and PYW files can only be set to one Python version. This is usually the latest installed Python version.  
- Environment variables are used by both Python versions. Their values, for example, for `PYTHONHOME`, must be set to the Python installation you want to work with. For an overview of the environment variables set by Python, refer to http://docs.python.org/2/using/cmdline.html. |
| Using dSPACE test automation with both Python versions in parallel | If your test automation scripts use dSPACE Python modules, distributed either via the dSPACE Python 2.5 setup or via the dSPACE Python Extensions setup available up to dSPACE Release 2013-A, and you do not want to migrate your scripts, you have to work with both Python versions. |
 Required Modifications in Your Scripts

Objective
Gives you information on modifications which you have to make in your scripts.

Where to go from here
Information in this section

| Migrating Changes in the Python Core Package | 42 |
| Migrating Changes in the PyWin32 Package | 45 |
| Using the dSPACE Library | 46 |
| Enhancements to the Standard Python 2.7 Distribution | 47 |

 Migrating Changes in the Python Core Package

Objective
When migrating your Python 2.5 applications to Python 2.7, note the following changes in the Python Core package.

Information in this topic

| Incompatible byte code on page 42 |
| Attribute from HierList class renamed on page 42 |
| Stricter evaluation of hexadecimal values on page 43 |
| Stricter check on argument list for listdir method on page 43 |
| Changed standard encoding on page 43 |
| Changed string representation of time structure on page 44 |

Incompatible byte code
The byte code of PYC files has changed. The interpreter of Python 2.7 does not start PYC files generated with Python 2.5.

- Recompile the PY files with Python 2.7.

Attribute from HierList class renamed
The list attribute of the HierList (object.object) class has been renamed to listControl. If the attribute is explicitly used in your source code or addressed via self._dict_, an exception is thrown because the attribute cannot be found.

- Change the list entries in your source code to listControl.
  or
- Implement a case construction to switch between the attribute names.
The `HierList` class is defined in `.Lib\site-packages\pythonwin\pywin\tools\hierlist.py`.

### Stricter evaluation of hexadecimal values

With Python 2.7, it is not sufficient to specify 0 as `0x`.
- Use `0x0` instead of `0x` to specify 0 in hexadecimal format.

To get the hexadecimal value of a string, you have to convert it to an integer value, for example, `int(HexStringValue, 16)`.

### Stricter check on argument list for listdir method

With Python 2.7, the `os.listdir(path)` method does not accept an empty string as an argument.
- Use `os.listdir('.')` instead of `os.listdir('')`.

### Changed standard encoding

With Python 2.7, the standard encoding is set to `ASCII` independently of the specified system encoding. This might lead to the following problems.

**Locale default string encoding**  If a script is parsed in Python 2.7 and no script encoding is declared, the Python standard default encoding, ASCII, is applied. If the script contains characters which are not ASCII characters, script parsing fails.

You can check the current setting with the following script:
```python
import locale
system_encoding = locale.getdefaultlocale()[1]
```
- You should always define script encoding to clarify script usage. The encoding has to be declared in the first or second line of the script.
  The following example declares the `latin-1` encoding.
  ```python
  # -*- coding: latin-1 -*-
  or
  
  # You can set the encoding globally by editing the setencoding function in \Lib\site.py.
  if 0:
      # Enable to support locale aware default string encodings.
      import locale
      loc = locale.getdefaultlocale()
      if loc[1]:
          encoding = loc[1]
  
  Replace if 0; by if 1;.
Migrating Python Scripts from Python 2.5 to Python 2.7

Information for AutomationDesk user

Python scripts used in AutomationDesk Exec blocks does not need an encoding declaration. If you do not specify an encoding declaration, the Python interpreter encodes the Exec block code as utf-8. Because, this has an affect on byte string literals, it is recommended to use always unicode literals for strings.

For further information on declaring encodings, refer to 2.1.4 Encoding declarations in the The Python Language Reference in the Python 2.7 documentation.

String conversion

If you encode a unicode string with a string conversion, for example, str(u'ä'), you get a UnicodeEncodeError exception when using it with Python 2.7.

- Use the encode method to cast a unicode string to a string, for example, u'ä'.encode('cp1252').

For further information on using unicode strings, refer to http://docs.python.org/2/howto/unicode.html.

Changed string representation of time structure

With Python 2.7, the string representation of the time structure returns a more descriptive string than with Python 2.5. Parsing the returned string will now fail.

Example:
repr(time.localtime())

Python 2.5 returns: '(2013,7,8,7,46,15,0,189,1)'

Python 2.7 returns: 'time.struct_time(tm_year=2013, tm_mon=7, tm_mday=8, tm_hour=7, tm_min=46, tm_sec=5, tm_wday=0, tm_yday=189, tm_isdst=1)'

- Add a tuple conversion to get the same return value.
  repr(tuple(time.localtime()))
# Migrating Changes in the PyWin32 Package

## Objective

When migrating your Python 2.5 applications to Python 2.7, note the following changes in the PyWin32 package.

## Information in this topic

- Removed manifest reference to MSVCR90 on page 45
- SAFEARRAY discontinued on page 45
- Bug in the Format method of pywintypes.Time on page 45
- Bug in the os.spawnv and os.spawnve functions on page 45

## Removed manifest reference to MSVCR90

With Python 2.7, pythoncom27.dll does not contain a manifest reference to MSVCR90.dll. If your script tries to access MSVCR90.dll, an exception is thrown. The calling application is now responsible for making MSVCR90.dll available.

- Copy MSVCR90.dll to the folder the calling application is stored in.

## SAFEARRAY discontinued

Calling the pythoncom.SAFEARRAY class leads to an error. This class is no longer supported.

- Use the new VARIANT class.

**Example:**

```python
# Python 2.5
pythoncom.SAFEARRAY([1, 2, 3, 4, 5], 1, pythoncom.VT_R8)

# Python 2.7
from win32com.client import VARIANT
VARIANT(pythoncom.VT_ARRAY | pythoncom.VT_R8, [1, 2, 3, 4, 5])
```

## Bug in the Format method of pywintypes.Time

The Format method of a pywintypes.Time object fails if the date returned by another COM method is earlier than 1900.

- To format the elements of the date (year, month, day), access them separately.

## Bug in the os.spawnv and os.spawnve functions

The os.spawnv and os.spawnve functions now do not accept an empty list for the process arguments.

- Provide at least one element in the list of arguments.
# Using the dSPACE Library

**Objective**
The Python 2.5 installation provided by dSPACE was already adapted to dSPACE-specific requirements. Some classes and methods are therefore not available in the Python 2.7 standard distribution. They are now provided by a dSPACE library that comes with the dSPACE software products that use Python. You only have to add an import command in your scripts to access this library.

## Information in this topic
- Using the Enums class on page 46
- Using the GetComIdentity method on page 46
- Using the IsTypeOf method on page 46

## Using the Enums class
The `win32com.Enums` class is not available in standard Python. It is provided by the dSPACE library.
- Add the following command to your scripts that use the `Enums` class:
  ```python
  from dspace.com import Enums
  ```

## Using the GetComIdentity method
The `GetComIdentity` method in `win32com.client` is not available in standard Python. It is provided by the dSPACE library.
- Add the following command to your scripts that use the `GetComIdentity` method:
  ```python
  from dspace.com import GetComIdentity
  ```

## Using the IsTypeOf method
The `IsTypeOf` method in `win32com.client` is not available in standard Python. It is provided by the dSPACE library.
- Add the following command to your scripts that use the `IsTypeOf` method:
  ```python
  from dspace.com import IsTypeOf
  ```
Enhancements to the Standard Python 2.7 Distribution

Objective

There are some dSPACE-specific enhancements to the standard Python 2.7. These either ensure the same behavior as before or solve known bugs.

Enhancements to get the same behavior as with Python 2.5

The following changes have been made to ensure that the Python installation behaves identically to the Python 2.5 dSPACE installation:

- 64-bit signed and unsigned integer data types can now be used in an automation call.
- Integer data types are now converted to valid ranges. Formerly, they were simply mapped to the double data type.
- Enhanced way to obtain error information after a failed automation call.
- The signed char data type is now handled as an array instead of converting it to an unsigned char data type.
- A PyOleNothing data type has been added to support interfaces which make use of the Visual Basic Nothing object.
- The Python Imaging Library (PIL) has been enhanced by a better dependency management to correct the wrongly referenced C-Runtime.
- By default, PY and PYW files are associated with Python.exe, which you can double-click to execute a file.
  The default behavior of Python 2.7 has been changed by dSPACE. The files are now opened in PythonWin.

Enhancements to solve known Python bugs

The following changes have been made to solve known bugs from Python 2.7:

- If an Item method was marked as the default property, the implementation for the index operator was not generated.
- The memory was not correctly freed if strings or interfaces were handled in arrays.
- The deadlock situation in multi-threaded scenarios using pythoncom has been solved.
- The default path for generated COM wrappers was not allowed to be located in a read-only location.
- The conversion of strings to CLSIDs which are created if a ProgId is given to the function was erroneous.
- The win32 API method FormatMessage that uses a lock internally lead to a multi-threading deadlock.
Python crashed if a COM object implemented in Python was released after the Python interpreter was shut down.

Python crashed if the Python windows hook for windows messages was still installed after the interpreter was shut down.

The generation of wrappers for standard COM interfaces was disabled if they were referenced in a type library.

Python still held a reference to an enumerator although it was reset in a COM server.

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

All changes to the PyWin32 package are reported in the bug tracking system, for example:

- https://sourceforge.net/p/pywin32/patches/123/
- https://sourceforge.net/p/pywin32/patches/127/
- https://sourceforge.net/p/pywin32/patches/128/
- https://sourceforge.net/p/pywin32/patches/130/
- https://sourceforge.net/p/pywin32/patches/609/
- https://sourceforge.net/p/pywin32/bugs/630/
- https://sourceforge.net/p/pywin32/bugs/638/
- https://sourceforge.net/p/pywin32/bugs/639/
- https://sourceforge.net/p/pywin32/bugs/640/
- https://sourceforge.net/p/pywin32/bugs/643/
- https://sourceforge.net/p/pywin32/bugs/649/
- https://sourceforge.net/p/pywin32/bugs/650/
- https://sourceforge.net/p/pywin32/bugs/651/
- https://sourceforge.net/p/pywin32/bugs/652/
- https://sourceforge.net/p/pywin32/bugs/653/

To identify the PyWin32 files changed by dSPACE the version number is changed from 218.0 to 218.10.
AutomationDesk

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>New Features of AutomationDesk 4.0</th>
<th>49</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrating to AutomationDesk 4.0</td>
<td>55</td>
</tr>
</tbody>
</table>

New Features of AutomationDesk 4.0

**General enhancements**

**Usability improvements**  The user interface of AutomationDesk has been switched to ribbon menus. The commands and features of AutomationDesk have been restructured to achieve a faster access to the frequently used functions. AutomationDesk also provides some new features.

- **Start page**
  
  Provides fast access to recently used projects, demo files and user documentation.

- **Favorites**
  
  Allows you to create an individual set of blocks and data objects from the built-in libraries and your own custom libraries.

- **Value Editor**
  
  Replaces the data object-specific editors for the standard data objects, like Int or Tuple. Especially, the features for editing tuples, lists and dictionaries have been improved.
- **Python Editor**
  The Python Editor used for `Exec` and `Eval` blocks now provides auto-completion and object-specific selection lists for methods and properties while you type a script.

- **Library context**
  You can now set a specific library context for your project or sequence. This allows you to temporarily resolve data object references in Custom Library templates without instantiating them.
  
  While the name of the referenced data object is stored at the data object, its value can be accessed only if the related context of a project or another Custom Library is available.

**Verdict handling**  The verdicts of an execution are now consistently handled in all libraries. Verdict-generating blocks from different libraries can now be used in the same sequence. The result state of a project contains one single verdict. Mixed result states like `Failed Undefined` no longer occur. Only the most severe verdict is reported. This facilitates the data exchange with SYNECT.

As a visible enhancement, the `Verdict` data object and the `GetVerdicts` automation block are now available in the Main Library. The `Decision` block has been adapted to the new verdicts. Internally, there are also modifications done in the Evaluation, Test Builder, Test Framework and the Framework Builder libraries. There is also a new `ResultState1` interface for the `ResultState` object in the AutomationDesk API that provides modified count properties, for example, the `PassedCount` property, which are adapted to the new verdict handling.

The naming and the severity of the verdicts are as follows:

- **Executed** (no result state symbol)
- **Passed**
- **Undefined**
- **Failed**
- **Error** (most severe verdict)

The `Fatal` verdict is now mapped to `Error`.

**VEOS support**  The VEOS support that was available as an add-on installation for AutomationDesk 3.6 is now included in AutomationDesk. It allows you to access VEOS 3.1 as an offline simulation platform.

**Synchronized platform access**  The platform management provides enhancements concerning the simultaneous access to the connected platforms via AutomationDesk and, for example, ControlDesk Next Generation.
Enhanced _INFO_ namespace  
The _INFO_ namespace now provides further properties to get information on a block via script:
- Author
- CreationDate and ModificationDate
- InstanceDescription and TemplateDescription
- Verdicts (new feature, see above)
For further information, refer to Getting Element Information (AutomationDesk Guide).

New automation libraries

**Platform Management library**  
With the Platform Management library, you can automate access to your simulation platform. For example, you can load a real-time application to your platform. For SCALEXIO and VEOS platforms, you can also start and stop the simulation during run time.
For further information, refer to Platform Management (AutomationDesk Library Reference).

**ModelDesk Access library**  
With the ModelDesk Access library, you can remote-control ModelDesk. It provides blocks to access projects and experiments, maneuvers, roads and traffic scenarios, for example. During run time, you can modify parameter values for testing the behavior with a changed parameter set.
For further information, refer to ModelDesk Access (AutomationDesk Library Reference).

Enhancements to the libraries

The following libraries have been enhanced:

**Main Library**  
The Main Library now provides a new data object:
- Verdict
  This data object is used to provide the result state of an execution.
  Its values are Executed, Passed, Undefined, Failed and Error.

The Main Library now provides two new basic elements:
- Invoke
  This block is used to invoke a block from a custom library or a built-in library by specifying its library link. The mechanism is similar to the dynamic link mode, but more flexible. You can parameterize the library link during run time, which allows you to react on current execution states by replacing the block to be executed.
GetVerdicts
This block is used to collect the verdicts in the specified block context and return the aggregated verdict. The block uses the new Verdict data object, see above. The block context can be modified by specifying a shift value, already known from the _INFO_ namespace.

The following block of the Main Library has been enhanced:

Decision
The block's Value data object has been replaced by the new Verdict data object. By this, the handling of verdicts and result states is now consistently managed. The block now also provides the Message data object to add a descriptive text to the resulting verdict.

For information on the block's migration aspects, refer to Migrating to AutomationDesk 4.0 on page 55.

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

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

DirectClearAllDTCs
This block is used to delete all trouble code entries from the fault memory.

DirectClearDTC
This block is used to delete a trouble code entry from the fault memory.

DirectReadDTCs
This block is used to read diagnostic trouble codes.

DirectReadDTCsWithEnvData
This block is used to read diagnostic trouble codes together with the environment data.

DirectReadEnvironmentData
This block is used to read the environment data of a specific diagnostic trouble code.

For further information, refer to ControlDesk NG Access (AutomationDesk Library Reference).
Dialogs library  The blocks of the Dialogs library have been enlarged by offline data objects. You can now specify output values to be used in the offline operation mode.

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

Framework Builder library  You can now add a DataContainer to a Frame block, an OperationSlot block or a SequenceFrame sequence.

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

HIL API Convenience library  The HIL API Convenience library provides a new data object:

- Vendor:

  Some blocks also have a Vendor data object. It specifies the vendor to identify the current implementation. By changing the vendor name, you can switch to a HIL API implementation of another vendor.

  The automation blocks are automatically migrated if you load an AutomationDesk project created with an earlier version. The default value is "dSPACE".

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

Remote Diagnostics (COM) library  The Remote Diagnostics (COM) library now supports Diagnostic Tool Set version 7.66.020 (Dts7.DSystem.7.66.020).

For further information, refer to Remote Diagnostics (COM) (AutomationDesk Library Reference).

Test Builder library  The Test Builder library was adapted to the new verdict handling. The AddVerdictForException block and the AddVerdict block provide Verdict data objects instead of String data objects. Internally, the Decision block of the Main Library is used to handle the verdicts. The new BaseBlock2 block is used to improve the internal verdict handling.

The configuration attributes for the Test Builder blocks has been enhanced by XML output and report formats.

For further information, refer to Test Builder (AutomationDesk Library Reference).
Enhancements to the COM API

The AutomationDesk COM API provides the following enhancements:

- The new `ResultState1` interface of the `ResultState` object provides properties to get the verdict of an execution. The `PassedCount`, `FailedCount` or `ErrorCount` property is set to 1 if the related verdict is returned by the executed sequence.

For further information, refer to `AutomationDesk API Reference`.

Starting AutomationDesk UI-free

If you start AutomationDesk without using the user interface (UI-free) via the AutomationDesk API, an AutomationDesk icon is placed in the notification area of the taskbar.

AutomationDesk - Automation Server has been discontinued as a separate executable.

For further information, refer to `Overview of the AutomationDesk API (AutomationDesk Guide)`.

Enhancements to report handling

There is a new report style sheet for `TestCase` sequences and standard sequences containing blocks of the Test Builder library. It is named `Detailed Report for TestBuilder`.

Enhancements to platform management

- **Enhanced platform support**  
  VEOS, the new offline simulation platform, is supported by AutomationDesk with the HIL API library.

- **Enhanced platform management functions**  
  AutomationDesk provides further commands for platform management. For example, you can update the firmware of MicroAutoBox or a modular system based on DS1005.

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

Discontinued libraries and blocks

With AutomationDesk 4.0, the following libraries are discontinued:

- ControlDesk Access
- Failure Simulation Access

The related Python modules `cdautomationlib` and `failuresimlib` are also discontinued. For further information on how to migrate, refer to `ControlDesk Next Generation Migration of ControlDesk 3.x Automation`.

---

New Features and Migration  November 2013
For information on migrating a project containing elements of a discontinued library, refer to Migrating to AutomationDesk 4.0 on page 55.

With AutomationDesk 4.0, the following block is discontinued:
- Platform Access – Stimulus

For further information on how to migrate, refer to Migrating to AutomationDesk 4.0 on page 55.

**Planned discontinuations**

With dSPACE Release 2014-A, the support of Microsoft® Visual SourceSafe will be discontinued as a version control system.

As AutomationDesk’s version control interface is based on the Microsoft Source Code Control (SCC) interface, you can still use Microsoft Visual SourceSafe, but it is not tested with AutomationDesk.

**Migrating to AutomationDesk 4.0**

**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.

Before you open an older project with the new AutomationDesk version, make sure that the following preconditions are fulfilled:
- You must create backups of the project and of the linked custom libraries.
- AutomationDesk must be running properly. There must not be any error messages displayed in the Log Viewer.
- The built-in libraries, required custom libraries and other packages must be loaded correctly.

If you are using a version control system, there are some preconditions for successful migration, refer to How to Migrate Projects Under Version Control (AutomationDesk Guide).
For further information, refer to *Migrating AutomationDesk* (AutomationDesk Guide).

### Main Library

**Modified Decision block**  The Decision block’s Value data object of the Int data type is replaced by the Verdict data object of Verdict data type. A reference used for the Value data object is preserved.

Existing block parameterizations are migrated as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Verdict</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 0</td>
<td>Undefined</td>
</tr>
<tr>
<td>= 0</td>
<td>Failed</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>Passed</td>
</tr>
</tbody>
</table>

The new Message data object is set to an empty string.

**Discontinued commands**  For compatibility reasons, the Add IfThenElse Condition command and the Delete IfThenElse Condition command were available for IfThenElse blocks migrated to AutomationDesk 2.0. These commands are now discontinued.

The Result element provided a command to export its contents to a file in PDF or HTML format. Because, a generated report for a result provides a more qualified output, this command is now discontinued.

### Other libraries

**HIL API library**  A parameterized BaseValue data object must be reinitialized by executing the related InitBaseValue block.

**HIL API Convenience library**  The data type of some data objects has changed from DataContainer to Variant. 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 data objects to be migrated manually.

---

You have to synchronize your sequences containing HIL API Convenience blocks with the HIL API Convenience library before using them.
Remote Diagnostics (COM) library  When you use the Remote Diagnostics (COM) library with ControlDesk 5.1 or later, accessing an ECU is possible only via the ControlDeskNG.D3system202 interface. You have to migrate AutomationDesk projects which use other diagnostic interfaces.

For further information, refer to Remote Diagnostics (COM) library (AutomationDesk Guide).

Test Builder library  The changed verdict handling requires enhancements for some block interfaces. Therefore, you have to synchronize your Test Case sequences and standard sequences containing Test Builder blocks with the Test Builder library before using them.

Automation Server  The AutomationDesk - Automation Server has been discontinued as a separate executable. With the Automation Server license, you cannot initialize the user interface of Automation Desk. You have to change the dispatch method in the Python code of your applications to migrate to AutomationDesk 4.0.

# Up to AutomationDesk 3.6
AudObj = win32com.client.Dispatch("ADAutomation.TAM")
# As of AutomationDesk 4.0
AudObj = win32com.client.Dispatch("AutomationDesk.TAM")

Discontinued libraries and blocks  If you open a project containing discontinued elements in AutomationDesk 4.0, the discontinued data objects are replaced by Discontinued data object data objects and the discontinued blocks are replaced by Discontinued block blocks during the automatic project update. This lets you load your projects and search for blocks and data objects to be migrated. If you execute a project containing elements of a discontinued library via AutomationDesk or API script, it will stop with an exception.

The following blocks and data objects are affected.

ControlDesk Access  The entire library is discontinued.

Affected blocks are:
- StartControlDesk
- ShowApplication
- LoadExperiment
- StartAnimationMode
- StopAnimationMode
- HideApplication
- CloseControlDesk

**Failure Simulation Access**  The entire library is discontinued.

Affected blocks are:
- ConnectFIU
- DisconnectFIU
- ResetFIU
- SetFailurePattern

The affected data object is:
- FailurePattern

**Platform Access**  The Stimulus block is discontinued because with dSPACE Release 2013-B, ControlDesk 3.x is discontinued. The Stimulus Editor and the admlib Python module are therefore no longer available.

Instead of the Stimulus block, you can use the Signal Editor of ControlDesk Next Generation and the stimulus feature of the HIL API library.

You can use the adm2stz converter. For download and instructions, refer to http://www.dspace.com/go/sdm2stz.

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

---

**User documentation renamed**  To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For AutomationDesk, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutomationDesk Reference</td>
<td>ADRef.pdf</td>
<td>AutomationDeskReference.pdf</td>
</tr>
<tr>
<td>AutomationDesk Library Reference</td>
<td>ADLibRef.pdf</td>
<td>AutomationDeskLibraryReference.pdf</td>
</tr>
<tr>
<td>AutomationDesk API Reference</td>
<td>ADAOAPIRef.pdf</td>
<td>AutomationDeskAPIReference.pdf</td>
</tr>
<tr>
<td>Document</td>
<td>Old Name</td>
<td>New Name</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
</tbody>
</table>
Automotive Simulation Models (ASM)

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>Blockset</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>All ASM Blocksets</td>
<td>62</td>
</tr>
<tr>
<td>ASM Base InCylinder Blockset</td>
<td>63</td>
</tr>
<tr>
<td>ASM Diesel Engine Blockset</td>
<td>65</td>
</tr>
<tr>
<td>ASM Diesel Exhaust Blockset</td>
<td>67</td>
</tr>
<tr>
<td>ASM Diesel InCylinder Blockset</td>
<td>70</td>
</tr>
<tr>
<td>ASM Drivetrain Basic Blockset</td>
<td>71</td>
</tr>
<tr>
<td>ASM Electric Components Blockset</td>
<td>72</td>
</tr>
<tr>
<td>ASM Environment Blockset</td>
<td>74</td>
</tr>
<tr>
<td>ASM Engine Gasoline Blockset</td>
<td>75</td>
</tr>
<tr>
<td>ASM Gasoline InCylinder Blockset</td>
<td>76</td>
</tr>
<tr>
<td>ASM Traffic Blockset</td>
<td>77</td>
</tr>
<tr>
<td>ASM Trailer Blockset</td>
<td>78</td>
</tr>
<tr>
<td>ASM Truck Blockset</td>
<td>80</td>
</tr>
<tr>
<td>ASM Turbocharger Blockset</td>
<td>82</td>
</tr>
<tr>
<td>ASM Vehicle Dynamics Blockset</td>
<td>84</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

### Migration of All ASM Blocksets

User documentation renamed

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For ASM, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASMDieselEngineModel Description</td>
<td>ASMModelDescriptionDiesel.pdf</td>
<td>ASMDieselEngineModelDescriptionDiesel.pdf</td>
</tr>
<tr>
<td>ASMDieselEngineInCylinderModelDescription</td>
<td>ASMModelDescriptionDieselInCylinder.pdf</td>
<td>ASMDieselEngineInCylinderModelDescription.pdf</td>
</tr>
<tr>
<td>ASMGasolineEngineBasicModelDescription</td>
<td>ASMModelDescriptionGasolineBasic.pdf</td>
<td>ASMGasolineEngineBasicModelDescription.pdf</td>
</tr>
<tr>
<td>ASMGasolineEngineModelDescription</td>
<td>ASMModelDescriptionGasoline.pdf</td>
<td>ASMGasolineEngineModelDescription.pdf</td>
</tr>
<tr>
<td>ASMGasolineEngineInCylinderModelDescription</td>
<td>ASMModelDescriptionGasolineInCylinder.pdf</td>
<td>ASMGasolineEngineInCylinderModelDescription.pdf</td>
</tr>
<tr>
<td>ASMTrailerModelDescription</td>
<td>ASMModelDescriptionTrailer.pdf</td>
<td>ASMTrailerModelDescription.pdf</td>
</tr>
<tr>
<td>ASMVehicleDynamicsModelDescription</td>
<td>ASMModelDescriptionVehicleDynamics.pdf</td>
<td>ASMVehicleDynamicsModelDescription.pdf</td>
</tr>
<tr>
<td>ASMVehicleDynamicsVEOSModelDescription</td>
<td>ASMModelDescriptionVehicleDynamicsVEOS.pdf</td>
<td>ASMVehicleDynamicsVEOSModelDescription.pdf</td>
</tr>
</tbody>
</table>
ASM Base InCylinder Blockset

Where to go from here

Information in this section

- New Features of ASM Base InCylinder Blockset 1.8
- Migrating to ASM Base InCylinder Blockset 1.8

New Features of ASM Base InCylinder Blockset 1.8

**TURBO_CONTROL block**

The controller has been extended by a feed forward control map which is based on engine speed and the desired induced engine torque.

**THROTTLE_VALVE block**

The block allows pressures to be zero, which is required to run engine parameterization variants with a single throttle on a configuration with multiple throttles.

**EGR_VALVE block**

The block allows pressures to be zero, which is required to run engine parameterization variants with a single EGR on a configuration with multiple EGR.

Migrating to ASM Base InCylinder Blockset 1.8

**INTAKE_MANIFOLD block**

The outports now include:
- Specific heat capacity at constant volume
- Specific heat capacity at constant pressure
- Isentropic ratio of gas in the intake manifold

**EXHAUST_MANIFOLD block**

The outports now include:
- Specific heat capacity at constant pressure
- Isentropic ratio of gas in the exhaust manifold

**CRANK_MECHANISM block**

The space in ASMSignalBus label has been removed. "V_Cyl[m3]" has been changed to "V_Cyl[m3]".
| SINGLEZONE_CYLINDER block | Temperature change (dT/dt) has been added to ASMSignalBus, which is required for pressure extrapolation. |
ASM Diesel Engine Blockset

Where to go from here

- New Features of ASM Diesel Engine Blockset 1.6.3
- Changes in the ASM Diesel Engine Demo Model
- Migrating to ASM Diesel Engine Blockset 1.6.3

New Features of ASM Diesel Engine Blockset 1.6.3

SOFT_ECU_DIESEL block

The SOFT_ECU subsystem Trq2quantity has been modified and renamed to SoftECU_FuelQuantity, so the Map_Trq2Quant parameter has been renamed to Map_q_Mean_Inj_Set.

The injection quantity is now calculated as the function of the indicated torque and engine speed instead of the torque only.

Two new parameters have been introduced:
- Const_q_Mean_Inj_Set[mm3/cyc]
- Sw_q_Mean_Inj[1Map/2Const]

Changes in the ASM Diesel Engine Demo Model

Engine subsystem

The Engine subsystem of ASM_EngineDiesel has been revised. There is a new Exhaust subsystem, which contains further subsystems: ExhaustSystem_SignalRouting and ExhaustSystem. ExhaustSystem contains the aftertreatment models and can be replaced by a desired exhaust system. To replace the exhaust system, drag and drop the ExhaustSystem from the exhaust library to the Exhaust subsystem.

The MVEM_Setup_Integrator subsystem with corresponding visibility tags has been moved from the ExhaustSystem subsystem one level up.
Migrating to ASM Diesel Engine Blockset 1.6.3

**COMMON_DIESEL_PARAMETERS block**

The units of Lower Heat Value and R Universal have been changed to SI units. The unit of Lower Heat Value is [J/kg] and the unit of R Universal is [J/(molK)]. Directly after the COMMON_DIESEL_PARAMETERS block, there is a Gain block to convert the unit of Lower Heat Value from [J/kg] to [kJ/kg]. After migration, R Universal is used in [J/(molK)].
ASM Diesel Exhaust Blockset

Where to go from here

Information in this section

| ASM Diesel Exhaust Blockset 1.3 | 67 |
| Changes in the ASM Diesel Exhaust Demo Model | 68 |
| Migrating to ASM Diesel Exhaust Blockset 1.3 | 69 |

ASM Diesel Exhaust Blockset 1.3

New Features of ASM_DieselExhaust blockset

The ASM_DieselExhaust blockset has been revised.

- The same ExhaustSystem can be used for ASM_InCylinder_EngineDiesel and ASM_EngineDiesel
- The block is vectorial, so it can simulate several ExhaustSystems simultaneously.

The following blocks are affected:

- DIESEL_OXIDATION_CATALYST
- DIESEL_PARTICULATE_FILTER
- PUMP_HOSE
- AIR_NON_RETURN_VALVE
- ADBLUE_PUMP
- UREA_DECOMPOSITION
- AIR_REGULATION_VALVE

DIESEL_OXIDATION_CATALYST block

New parameter: Const_num_DOC

DIESEL_PARTICULATE_FILTER block

New parameter: Const_num_DPF

The units of R_Universal have changed from [kJ/(molK)] to [J/(molK)] and of Const_Ep_Soot from [kJ/mol] to [J/mol].

PUMP_HOSE block

New parameter: Const_num_PumpHose

AIR_NON_RETURN_VALVE block

New parameter: Const_num_AirNonRetValve
ADBLUE_PUMP block

New parameter: Const_num_AdBluePump

UREA_DECOMPOSITION block

New inports:
- Const_cp_AdBlue_[J/(kgK)]
- cp_Air_[J/(kgK)]
- cp_Exhaust_[J/(kgK)]

AIR_REGULATION_VALVE block

New parameter: Const_num_AirRegValve

COMMON_EXHAUST_PARAMETERS block

New parameter: Const_M_Air

New inports:
- R_ExhMan_[J/(kgK)]
- R_Universal_[J/kmo]
- R_Air_[J/(kgK)]
- cp_Air_[J/(kgK)]
- kappa_Air[]

Changes in the ASM Diesel Exhaust Demo Model

ASM_DieselExhaust_lib/Demos

The new demo model consists of three blocks:
- ExhaustSystem_DOC_DPF
- ExhaustSystem_DOC_DPF_SCR
- ExhaustSystem_DOC_DPF_NonAir_SCR

ExhaustSystem_DOC_DPF

The number and some names of inports have changed. The order of the inports has changed. The new inports are:
- cp_ExhMan_[J/(kgK)]
- R_ExhMan_[J/(kgK)]
- cv_ExhMan_[J/(kgK)]
- IO_SCR_Supply[0_1]
- mdot_Gas_[kg/s], which stands for mdot_Out_EGR_[kg/h] and mdot_Out_Engine_noPost_[kg/h]
- T_EngineCoolant_[degC] has been renamed to T_Water_[degC]
The Setup_Integrator block has been moved a level up and has been renamed in ASM_EngineDiesel to MVEM_Setup_Integrator and in InCylinder_EngineDiesel to InCyl_Setup_Integrator.

The Common_Exhaust_Parameters block has been modified. It has a new inport:
- R_ExhMan[J/(kg*K)]

The Common_Exhaust_Parameters block has four GoToFrom connections:
- Const_Rm[J/(mol*K)]
- R_Air[J/(kg*K)]
- cp_Air[J/(kg*K)]
- kappa_Air[]

The units of the GoToFrom connections lower heat value of fuel[J/kg] and R_Universal[J/(mol*K)] have been changed to [J/kg] and [J/(mol*K)], respectively.

For inports, units of the GoToFroms, Setup_Integrator and Common_Exhaust_Parameters blocks apply the same changes as for ExhaustSystem_DOC_DPF. In addition, any virtual GoToFrom connection has been replaced by lines.

SCR supply system has been modified. The control signals for the supply system are routed into system by an inport instead of by the GoToFrom connection. A new TAG_SCR_SupplySystem inport has been introduced here to route the integrator signals into system.

Here, the same changes apply as for ExhaustSystem_DOC_DPF_SCR.

The unit of Const_Ep_Soot has been changed to the SI unit [J/mol].

Two outports of the block are renamed. The Exhaust_SCR_Cell_SignalBus outport has been renamed to SCR_Cat_SignalBus, the Exhaust_SCR_SupplySystem_SignalBus outport has been renamed to SCR_SupplySystem_SignalBus.
ASM Diesel InCylinder Blockset

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changes in the ASM Diesel InCylinder Demo Model</td>
</tr>
<tr>
<td>Migrating to ASM Diesel InCylinder Blockset 1.7</td>
</tr>
</tbody>
</table>

Changes in the ASM Diesel InCylinder Demo Model

Engine

The Engine subsystem of ASM_inCylinderDiesel has been revised. A new Exhaust subsystem, which contains further Massflow and ExhaustSystem subsystems, has been introduced. The InCylinderDiesel model can now simulate an SCR system.

Migrating to ASM Diesel InCylinder Blockset 1.7

COMMON_INCYLINDER_DIESEL_PARAMETERS block

The unit of the Const_Rm parameter has been changed from [J/(kmol K)] to [J/(mol K)].
ASM Drivetrain Basic Blockset

Migrating to ASM Drivetrain Basic Blockset 2.0.1

| SOFT_ECU_TRANSMISSION_BASIC block | The FurtherShiftTables ground blocks have been replaced by constant zero blocks to avoid dimension problems in Simulink. |
## ASM Electric Components Blockset

| New Features of ASM Electric Components Blockset 2.5 |  
|---------------------------------------------------|---|
| SQUIRREL_CAGEASYNCHRONOUS_MACHINE_D_Q block | New signals (e.g., power signals) have been added to the ASMSignalBus. |
| PERMANENT_MAGNETSYNCHRONOUS_MACHINE_D_Q block | New signals (e.g., power signals) have been added to the ASMSignalBus. The back EMV voltage calculation has been expanded with the voltage inducted from id and iq. |
| BRUSHLESS_DCMACHINE_ALPHA_BETA block | New signals (e.g., power signals) have been added to the ASMSignalBus. |
| SEPARATELY_EXCITED_DC_MACHINE block | New signals (e.g., power signals) have been added to the ASMSignalBus. |
| PMSM_D_Q_NONLINEAR block | New signals (e.g., power signals) have been added to the ASMSignalBus. The back EMV voltage calculation has been expanded with the voltage inducted from id and iq. |
| THREE_PHASE_INVERTER block | New signals (e.g., power signals) have been added to the ASMSignalBus. The inverter behavior if both gates (HSD and LSD) are on or off at the same time has been improved. |
| PMSM_CONTROLLER block | A reset of the space vector modulator has been added. |
| PMSM_CONTROLLER_THREE_LEVEL block | A reset of the three level space vector modulator has been added. |
Automotive Simulation Models (ASM)

BLDC_CONTROLLER block
A reset of the block modulator has been added.

BLOCK_MODULATOR block
A reset inport has been added.

SPACE_VECTOR_MODULATOR block
A reset inport has been added.

THREE_LEVEL_SPACE_VECTOR_MODULATOR block
A reset inport has been added.

HALF_BRIDGE_INVERTER block
New signals (e.g., power signals) have been added to the ASMSignalBus. The inverter behavior if both gates (HSD and LSD) are on or off at the same time has been improved.

BATTERY block
New signals (e.g., power signals) have been added to the ASMSignalBus.

BATTERY_MULTICELL block
New signals (e.g., power signals) have been added to the ASMSignalBus.

PARALLEL_CIRCUIT_CURRENT_SPLIT block
New outport for the terminal voltage of the complete battery parallel circuit has been added.

DC_LINK block
This new block presents a DC link circuit. It is used as an intermediate circuit to connect the power supply (e.g., battery) with the load system (e.g., inverter or passive load).

ENGINE_INERTIA block
An inport for reset and the initial engine speed has been added.

Changes in the ASM Electric Components Demo Model

Permanent magnet synchronous machine d q
To connect the battery and the inverter, a DC link circuit has been added to the ASM_ElectricDrive_PMSM_TwoLevelInverter demo model.
ASM Environment Blockset

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>New Features of ASM Environment Blockset 3.1</th>
<th>74</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrating to ASM Environment Blockset 3.1</td>
<td>74</td>
</tr>
</tbody>
</table>

New Features of ASM Environment Blockset 3.1

CONTROLLER block
A reset inport has been added.

GEAR_SHIFTER block
A reset inport has been added.
The simulation step size has been added as parameter.

LATERAL_CONTROL1 block
A reset inport has been added.

LATERAL_CONTROL2 block
A reset inport has been added.

V_ROAD_REF block
A reset inport has been added.
The simulation step size has been added as parameter

Migrating to ASM Environment Blockset 3.1

Reset functionality
Several blocks have a new reset inport. During migration of the model, the new inports are connected to Constant blocks and therefore the reset functionality cannot be used yet. If the reset functionality is required, the inport must be connected to the reset output from the maneuver scheduler block that is available on the ASM signal bus.
ASM Engine Gasoline Blockset

New Features of ASM Engine Gasoline Blockset 2.4.3

<table>
<thead>
<tr>
<th>SOFT_ECU_GASOLINE</th>
<th>A new function has been introduced in SoftECU to calculate the Lambda_Set value. The calculation is based on a look-up table. This function contains three new parameters:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>■ Const_Lambda_Set</td>
</tr>
<tr>
<td></td>
<td>■ Map_Lambda_Set</td>
</tr>
<tr>
<td></td>
<td>■ Sw_Lambda_Set</td>
</tr>
<tr>
<td></td>
<td>The Map_Teq2RelAirmass parameter now depends on engine speed. It has therefore been renamed to Map_RelAirmass_Set.</td>
</tr>
</tbody>
</table>
|                   | The Map_phi_ign_opt parameter can be calculated from the measurement data or from a universal look-up table. A switch is available in the SoftECU settings of ASMParame...

| COMBUSTION_TORQUE_SI | The Map_phi_ign_opt parameter can be calculated from measurement data or a universal map. A switch is available in SoftECU settings of ASMParame...


ASMAutoMobile Simulation Models (ASM)

New Features and Migration November 2013

ASM Gasoline InCylinder Blockset

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of ASM Gasoline InCylinder Blockset 1.7</td>
</tr>
<tr>
<td>Migrating to ASM Gasoline InCylinder Blockset 1.7</td>
</tr>
</tbody>
</table>

New Features of ASM Gasoline InCylinder Blockset 1.7

PORTINJECTOR_CONTINUOUS block

The offset time can now be specified by a map which depends on the battery voltage.

Migrating to ASM Gasoline InCylinder Blockset 1.7

PORTINJECTOR_CONTINUOUS block

The migration process creates a fixed offset time map based on the original offset time variable.

The dimension issue due to a number of injections unequal to one has been solved.
ASM Traffic Blockset

Changes in the ASM Traffic Demo Model

| Reset functionality | Now several blocks in the environment, drivetrain and soft ECU subsystems have a reset input that is triggered by the maneuver reset. |
## ASM Trailer Blockset

<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 ASM Trailer Blockset 2.2</strong></td>
<td>78</td>
</tr>
<tr>
<td><strong>Changes in the ASM Trailer Demo Model</strong></td>
<td>79</td>
</tr>
<tr>
<td><strong>Migrating to ASM Trailer Blockset 2.2</strong></td>
<td>79</td>
</tr>
</tbody>
</table>

### New Features of ASM Trailer Blockset 2.2

<table>
<thead>
<tr>
<th>SUSKIN_RIGID_SYM block</th>
<th>With this new suspension kinematics model, a rigid axle is modeled and provides a more convenient parameterization than the SUSKIN_SYM_3DOF model.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUSCOMP_RIGID_SYM block</td>
<td>This new suspension compliance model is added for a rigid axle. With this new model a compliance effect of a rigid axle can be parameterized.</td>
</tr>
<tr>
<td>CONTACT_POINT_CALCULATION_xyz block</td>
<td>The contact point calculation now takes the tire camber angle into account.</td>
</tr>
<tr>
<td>TIRE_MODEL_TMEASY_xyz block</td>
<td>The friction coefficient of the tire force calculation at standstill has been improved.</td>
</tr>
<tr>
<td>TIRE_MODEL_MAGIC_FORMULA_xyz block</td>
<td>The tire force calculation at standstill takes the friction coefficient into account.</td>
</tr>
</tbody>
</table>
Changes in the ASM Trailer Demo Model

Reset functionality
Now several blocks in the environment, drivetrain and soft ECU subsystems have a reset import that is triggered by the maneuver reset.

Migrating to ASM Trailer Blockset 2.2

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTACT_POINT_CALCULATION_TRAILER_xxx</td>
<td>The \texttt{Angle_Alpha_Tire[rad]} output has been renamed to \texttt{Sin_Angle_Alpha_Tire[]}. The bus signal label has been changed from \texttt{Angle_Alpha_xxx_Tire[deg]} to \texttt{Sin_Angle_Alpha_xxx_Tire[]} and the trigonometry calculation \texttt{asin} has been removed.</td>
</tr>
<tr>
<td>TIRE_MODEL_TMEASY_TRAILER_xxx</td>
<td>The import name has been changed from \texttt{Angle_Alpha_Tire[rad]} to \texttt{Sin_Angle_Alpha_Tire[]}, as \texttt{\texttt{sin(angle_alpha)}} is used instead of \texttt{angle_alpha[rad]}.</td>
</tr>
<tr>
<td>TIRE_MODEL_MAGIC_FORMULA_TRAILER_xxx</td>
<td>The import name has been changed from \texttt{Angle_Alpha_Tire[rad]} to \texttt{Sin_Angle_Alpha_Tire[]}. The trigonometry sinus calculation has been removed.</td>
</tr>
<tr>
<td>TIRE_MODEL_TMEASY_DUMMY_xxx</td>
<td>The import name has been changed from \texttt{Angle_Alpha_Tire[rad]} to \texttt{Sin_Angle_Alpha_Tire[]}.</td>
</tr>
<tr>
<td>TIRE_MODEL_MAGIC_FORMULA_DUMMY_xxx</td>
<td>The import name has been changed from \texttt{Angle_Alpha_Tire[rad]} to \texttt{Sin_Angle_Alpha_Tire[]}.</td>
</tr>
<tr>
<td>SUSPENSION_COMPLIANCE_xxx</td>
<td>The new \texttt{Pos_Wheel[Left;Right]} import has been added. This import is used only for the new Suspension Compliance Rigid Axle Symmetric.</td>
</tr>
<tr>
<td>SUSPENSION_COMPLIANCE_OPP_xxx</td>
<td>The new \texttt{Pos_Wheel[Left;Right]} import has been added. This import is used only for the new Suspension Compliance Rigid Axle Symmetric.</td>
</tr>
</tbody>
</table>
ASM Truck Blockset

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of ASM Truck Blockset 2.1</td>
<td></td>
</tr>
<tr>
<td>Changes in the ASM Truck Demo Model</td>
<td></td>
</tr>
<tr>
<td>Migrating to ASM Truck Blockset 2.1</td>
<td></td>
</tr>
</tbody>
</table>

New Features of ASM Truck Blockset 2.1

SUSKIN_RIGID_SYM block

With this new suspension kinematics model, a rigid axle is modeled and provides a more convenient parameterization than the SUSKIN_SYM_3DOF model.

SUSCOMP_RIGID_SYM block

The new suspension compliance model has been added for a rigid axle. With this new model a compliance effect of a rigid axle can be parameterized.

CONTACT_POINT_CALCULATION_xxx block

The contact point calculation now takes the tire camber angle into account.

TIRE_MODEL_TMEASY_xxx block

The tire force calculation at standstill has been improved by taking the friction coefficient into account.

TIRE_MODEL_MAGIC_FORMULA_xxx block

The tire force calculation at standstill takes the friction coefficient into account.
Changes in the ASM Truck Demo Model

Reset functionality
Now several blocks in the environment, drivetrain and soft ECU subsystems have a reset import that is triggered by the maneuver reset.

Migrating to ASM Truck Blockset 2.1

<table>
<thead>
<tr>
<th>BLOCK NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTACT_POINT_CALCULATION_xxx</td>
<td>The Angle_Alpha_Tire[rad] outport has been renamed to Sin.Angle_Alpha_Tire[]. The bus signal label has been changed from Angle_Alpha_xxx_Tire[deg] to Sin.Angle_Alpha_xxx_Tire[] and the trigonometry calculation asin has been removed.</td>
</tr>
<tr>
<td>TIRE_MODEL_TMEASY_xxx</td>
<td>The inport name has been changed from Angle_Alpha_Tire[rad] to Sin.Angle_Alpha_Tire[]. The block uses sin(angle_alpha) instead of angle_alpha[rad].</td>
</tr>
<tr>
<td>TIRE_MODEL_MAGIC_FORMULA_xxx</td>
<td>The inport name has been changed from Angle_Alpha_Tire[rad] to Sin.Angle_Alpha_Tire[]. The trigonometry sin calculation has been removed.</td>
</tr>
<tr>
<td>TIRE_MODEL_TMEASY_DUMMY_xxx</td>
<td>The inport name has been changed from Angle_Alpha_Tire[rad] to Sin.Angle_Alpha_Tire[].</td>
</tr>
<tr>
<td>TIRE_MODEL_MAGIC_FORMULA_DUMMY_xxx</td>
<td>The inport name has been changed from Angle_Alpha_Tire[rad] to Sin.Angle_Alpha_Tire[].</td>
</tr>
<tr>
<td>SUSPENSION_COMPLIANCE_xxx</td>
<td>The new Pos.Wheel[Left;Right] inport has been added. This inport will be used only for the new Suspension Compliance Rigid Axle Symmetric.</td>
</tr>
<tr>
<td>SUSPENSION_COMPLIANCE_OPP_xxx</td>
<td>The new Pos.Wheel[Left;Right] inport has been added. This inport will be used only for the new Suspension Compliance Rigid Axle Symmetric.</td>
</tr>
</tbody>
</table>
ASM Turbocharger Blockset

Where to go from here

Information in this section

- New Features of ASM Turbocharger Blockset 2.1.2
- Migrating to ASM Turbocharger Blockset 2.1.2

New Features of ASM Turbocharger Blockset 2.1.2

Improved support of vectorial engine configurations

It is now possible to run parameter variants with a single turbocharger system on engine configurations with multiple turbocharger systems. The issues due to zero input values are solved. The following blocks are affected:

- COMPRESSOR
- COMPRESSOR_HP
- TURBINE
- TURBINE_SAEJ922
- TURBINE_HP
- WASTEGATE_VALVE
- WASTEGATE_VALVE_HP

Migrating to ASM Turbocharger Blockset 2.1.2

Turbocharger

The turbine now has a specific heat capacity at constant pressure and an isentropic ratio of gas as input ports instead of Goto-From connections. This also applies to the high-pressure turbine, which will use values calculated in the post turbine HP manifold.

POSTTURBHPMAN

The block now calculates the specific heat capacity at constant pressure and the isentropic ratio of gas in the post turbine HP manifold. These values are available as outports.

COMPRESSOR

The block has been changed internally to support zero input values.

COMPRESSOR_HP

The block has been changed internally to support zero input values.
<table>
<thead>
<tr>
<th>Block Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TURBINE</td>
<td>The block has been changed internally to support zero input values.</td>
</tr>
<tr>
<td>TURBINE_SAEJ922</td>
<td>The block has been changed internally to support zero input values.</td>
</tr>
<tr>
<td>TURBINE_HP</td>
<td>The block has been changed internally to support zero input values.</td>
</tr>
<tr>
<td>WASTEGATE_VALVE</td>
<td>The block has been changed internally to support zero input values.</td>
</tr>
<tr>
<td>WASTEGATE_VALVE_HP</td>
<td>The block has been changed internally to support zero input values.</td>
</tr>
</tbody>
</table>
ASM Vehicle Dynamics Blockset

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>New Features of ASM Vehicle Dynamics Blockset 2.5</em></td>
</tr>
<tr>
<td><em>Changes in the ASM Vehicle Dynamics Demo Model</em></td>
</tr>
<tr>
<td><em>Migrating to ASM Vehicle Dynamics Blockset 2.5</em></td>
</tr>
</tbody>
</table>

New Features of ASM Vehicle Dynamics Blockset 2.5

**SUSKIN_RIGID_SYM block**

With this new suspension kinematics model, a rigid axle is modeled that provides more convenient parameterization than the SUSKIN_SYM_3DOF model.

**SUSCOMP_RIGID_SYM block**

A new suspension compliance model has been added for a rigid axle. With this new model you can parameterize a compliance effect of a rigid axle.

**CONTACT_POINT_CALCULATION_xxx block**

The contact point calculation considers the tire camber angle.

**TIRE_MODEL_TMEASY_xxx block**

The tire force calculation at standstill considers the friction coefficient.

**TIRE_MODEL_MAGIC_FORMULA_xxx block**

The tire force calculation at standstill considers the friction coefficient.

**IDLE_SPEED_CONTROL_ENGINE_BASIC block**

A reset inport has been added.

**SOFT_ECU_TRANSMISSION block**

A reset inport has been added. The simulation step size has been added as a parameter.
Changes in the ASM Vehicle Dynamics Demo Model

**Reset functionality**
Several blocks in the environment, drivetrain and soft ECU subsystems now have a reset import that is triggered by the maneuver reset.

**Migrating to ASM Vehicle Dynamics Blockset 2.5**

**STEERING_VARIABLE_RATIO block**
The dimensions of the Map_i_Steering and Map_phi_SteeringColumn look-up tables have been increased from [25,1] to [60,1]. These extensions are done automatically during initialization via a call of the steering_postmigrate variant, for example, in the go function.

**CONTACT_POINT_CALCULATION_xxx block**
The output Angle_Alpha_Tire[rad] has been renamed to Sin_Angle_Alpha_Tire[]. The bus signal label has been changed from Angle_Alpha_xxx_Tire[deg] to Sin_Angle_Alpha_xxx_Tire[] and the trigonometry calculation asin has been removed.

**TIRE_MODEL_TMEASY_xxx block**
The inport name has been changed from Angle_Alpha_Tire[rad] to Sin_Angle_Alpha_Tire[], as sin(angle_alpha) is used instead of angle_alpha[rad].

**TIRE_MODEL_MAGIC_FORMULA_xxx block**
The inport name has been changed from Angle_Alpha_Tire[rad] to Sin_Angle_Alpha_Tire[]. The trigonometry sin calculation has been removed.

**TIRE_MODEL_TMEASY_DUMMY_xxx block**
The inport name has been changed from Angle_Alpha_Tire[rad] to Sin_Angle_Alpha_Tire[].

**TIRE_MODEL_MAGIC_FORMULA_DUMMY_xxx block**
The inport name has been changed from Angle_Alpha_Tire[rad] to Sin_Angle_Alpha_Tire[].

**SUSPENSION_COMPLIANCE_xxx block**
A new Pos_Wheel[Left;Right] inport has been added. This inport will be used only for the new Suspension Compliance Rigid Axle Symmetric.

**SUSPENSION_COMPLIANCE_OPP_xxx block**
A new Pos_Wheel[Left;Right] inport has been added. This inport will be used only for the new Suspension Compliance Rigid Axle Symmetric.
### New Features and Migration November 2013

<table>
<thead>
<tr>
<th>SOFT_ECU_TRANSMISSION block</th>
<th>The <code>FurtherShiftTables</code> ground blocks have been replaced by constant zero blocks to avoid dimension problems in Simulink.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRANK_SHAFT block</td>
<td>The lower saturation limit of the integrator has been changed to &quot;-inf&quot;. This makes the block compatible with virtual vehicle systems.</td>
</tr>
</tbody>
</table>
ConfigurationDesk

ConfigurationDesk – Implementation

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of ConfigurationDesk 5.0</td>
<td>87</td>
</tr>
<tr>
<td>Migrating to ConfigurationDesk 5.0</td>
<td>92</td>
</tr>
</tbody>
</table>

New Features of ConfigurationDesk 5.0

More intuitive user interface

ConfigurationDesk’s user interface is more intuitive: Its menu bar and toolbars have been replaced by *ribbons* and the *Backstage view* used in Microsoft Office, etc. For details, refer to *Basics on Ribbons* (*ConfigurationDesk Real-Time Implementation Guide*).

Embedded help for Conflicts Viewer

The Conflicts Viewer now offers a text field containing useful information on a selected conflict as well as ways to resolve the conflict (if required).
Excel export files

- The format of the exported device topology and external cable harness Excel files have been updated. For details, refer to Importing a Device Topology from a Microsoft Excel™ Sheet (ConfigurationDesk Real-Time Implementation Guide) or Description of an Exported Cable Harness Microsoft Excel™ Sheet (ConfigurationDesk Real-Time Implementation Guide).
- The complete configuration data of a ConfigurationDesk application can now be exported to an Excel file. For instructions, refer to How to Export the Configuration of an Application (ConfigurationDesk Real-Time Implementation Guide).

Support of Simulink bus signals

ConfigurationDesk supports Simulink bus signals as input/output signals of Data Inport/Data Outport blocks in a Simulink model. For details, refer to Basics of Data Port Blocks with Structured Data Ports (ConfigurationDesk Simulink Modeling Guide).

Flatter structures for executable applications

ConfigurationDesk now provides a clearer structure in the Executable Application table view. For a flatter hierarchy, runnable functions and events are no longer grouped in folders, but displayed directly beneath the Components folder or beneath the tasks they are assigned to. The type of each element is described in the Element type property. For details on the structure of executable applications, refer to Basics on Modeling Executable Applications (ConfigurationDesk Real-Time Implementation Guide).

Assigning executable application elements via context menu commands

In the Executable Application table view, you can now assign elements to other elements or components via context menu commands, for example, you can assign an event to a task via the task’s context menu. Refer to How to Model an Executable Application Manually From Scratch (ConfigurationDesk Real-Time Implementation Guide).

New Task Configuration table view

ConfigurationDesk now provides the Task Configuration table view, which gives you a clear overview of all the tasks and task groups it contains. You can inspect and configure the properties of the tasks and task groups. This is especially useful for large multicore applications. Refer to Task Configuration Table View (ConfigurationDesk Real-Time Implementation Reference).
Synchronized communication between SCALEXIO systems via Gigalink

ConfigurationDesk now supports event-based synchronization of applications running on two or more SCALEXIO systems via Gigalink connection. In addition, you can specify synchronized reading of data by using the blocking or non-blocking mode of the Gigalink function block.

For details, refer to Basics on Gigalink Communication (ConfigurationDesk Real-Time Implementation Guide).

V-ECU implementation enhancements

ConfigurationDesk provides the following enhancements for V-ECU implementations:

- **Additional AUTOSAR OS elements for AUTOSAR OS tasks**
  - ConfigurationDesk displays the following additional AUTOSAR OS task properties in the Properties Browser:
    - AUTOSAR OS resources
    - AUTOSAR OS events
  - Refer to Configuring Properties of Tasks and Events (ConfigurationDesk Real-Time Implementation Guide).

- **Additional AUTOSAR OS task information variables in the A2L file**
  - A2L files generated for application processes that have V-ECU implementations assigned now contain the following additional task information variables:
    - `<TaskName>_TaskCallCount`
    - `<TaskName>_OverrunCount`
    - `<TaskName>_TurnaroundTime`

- **Update path of V-ECU implementations**
  - If you update a V-ECU implementation, ConfigurationDesk tries to find the V-ECU implementation container at a file location relative to the ConfigurationDesk application's root directory. If that fails, ConfigurationDesk tries to find the V-ECU implementation container at the file location it was originally imported from. Refer to Updating V-ECU Implementations Used in a ConfigurationDesk Application (ConfigurationDesk Real-Time Implementation Guide).
New function block types

With ConfigurationDesk 5.0, the following function block types are available:

- **Waveform Current Sink**
  The Waveform Current Sink function block lets you simulate sensors with a current interface, like Hall sensors. The load can be changed at run time without the concurrence of the behavior model, for example, to produce cyclic current profiles.

- **Waveform Voltage Out**
  The Waveform Voltage Out function block can be used to generate periodic voltage output signals, for example, to simulate signals generated by combined oil temperature and pressure sensors in combustion engines.

- **Waveform Digital Out**
  The Waveform Digital Out function block can be used to generate periodic pulses, for example, to simulate low-speed data packages (protocols).

- **Current Signal Capture**
  With the Current Signal Capture function block, you can measure analog current signals (coming from an external device, e.g., ECU) by capturing signal sequences at configurable sample rates.

- **Voltage Signal Capture**
  With the Voltage Signal Capture function block, you can measure analog voltage signals (coming from an external device, e.g., ECU) by capturing signal sequences at configurable sample rates.

Enhanced function block types

**CAN function block**

The CAN function block now lets you activate bus statistics. If enabled, the model interface of the CAN function block provides the Bus Statistic function with several bus statistics ports for data exchange between the I/O model and the behavior model. Bus statistic information includes general frame-specific and error-specific information. You can specify how often the bus statistics functions are executed.

**Gigalink function block**

The Gigalink function block now provides features to support synchronized communication between SCALEXIO systems. For example, you can specify synchronized reading of data by using the blocking or non-blocking mode.
**SENT Out function block**  The SENT Out function block provides the following new features according to the SAE J2716 JAN2010 SENT standard:

- Transmitting serial messages (in short and enhanced serial message format) via a defined number of SENT messages.
- Using predefined data protocols for data transmission to support specific sensor applications.
- The pause pulse feature has been enhanced: Now the pause value also can be provided by the behavior model regardless of the SENT message data content.

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

<table>
<thead>
<tr>
<th>New Function Custom Properties table view</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfigurationDesk now provides the Function Custom Properties table view to access and configure specific properties of custom function blocks via table. Refer to <em>Function Custom Properties Table View</em> (<a href="#">ConfigurationDesk Real-Time Implementation Reference</a>).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Limitations concerning MATLAB compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>The following limitations apply when you work with MATLAB R2012a and R2013b.</td>
</tr>
</tbody>
</table>

**Problems with spaces in application path names when using MATLAB R2012a**  Due to problems with spaces in application path names, it is recommended to install the following bug fix from the MathWorks® website before using ConfigurationDesk with MATLAB R2012a: http://www.mathworks.com/support/bugreports/814551

**Restricted access to blocksets when using MATLAB R2012a**  If you use MATLAB R2012a, dSPACE blocksets from dSPACE Release 2013-B are not included in the MATLAB Start menu.

You can still access the blocksets from the Simulink Library Browser or by typing the library name in the MATLAB Command Window. The MATLAB versions R2012b and later do not provide the Start menu.

**No support of UTF-16 encoding for signal and block names when using MATLAB R2013b**  Block names, signal labels and annotations using the UTF-16 encoding are not supported by ConfigurationDesk. Such characters might lead to problems when you generate code.
Migrating to ConfigurationDesk 5.0

With ConfigurationDesk 4.4, a project-specific search path for custom functions was added. Projects using this search path do not require any adjustments.

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

- Copy the XML file of each custom function type and, if available, the header file (`<Function_block_type_name>/CModule_name>.h`), the C++ source code file (`<Function_block_type_name>/CModule_name>.cpp`), and the type definition file (`<Function_block_type_name>/CModule_name>_TypeDef.h`) either to the project-specific search path or to the global search path.

  The project-specific custom function directory is:

  `<DocumentsFolder>/<Project>/CustomFunctions`

  The default global search path for custom functions is:

  `<InstallationFolder>/ConfigurationDesk/Implementation/UserFiles`

  Since this directory is normally read-only, you should change the path in the `Settings` dialog, which you can open via `File – Settings` (see `Settings (ConfigurationDesk Real-Time Implementation Reference)`).

  Afterwards, the function block type can be resolved via `Reload Definition` from its context menu.

- Change the global search path for custom functions to a folder containing the missing XML files. The corresponding custom function types are resolved automatically.

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

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

If you work with the SYNECT database, ConfigurationDesk 5.0 supports only SYNECT 1.2. You have to migrate a SYNECT database with ConfigurationDesk data from a previous SYNECT database version.

SYNECT server provides a central database. Migrating SYNECT’s database therefore affects all of the client users that connect to SYNECT server. This means you have to migrate the database in a central process.

For further information on migrating to SYNECT server 1.2, refer to Migrating to SYNECT Server 1.2 (SYNECT Server Document).

User documentation renamed

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For ConfigurationDesk Implementation Version, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started with SCALEXIO</td>
<td>CfgDeskSCALEXIOGetting Started.pdf</td>
<td>ConfigurationDeskGetting Started WithSCALEXIO.pdf</td>
</tr>
<tr>
<td>ConfigurationDesk Tutorial</td>
<td>CfgDeskImplTutorial.pdf</td>
<td>ConfigurationDeskTutorial.pdf</td>
</tr>
<tr>
<td>ConfigurationDesk Automating Tool Handling</td>
<td>ConfigurationDesk AutomatingToolHandling .pdf</td>
<td>ConfigurationDeskAutomatingTool Handling.pdf</td>
</tr>
<tr>
<td>Document</td>
<td>Old Name</td>
<td>New Name</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>ConfigurationDesk UART API Reference</td>
<td>CfgDeskUARTAPIRef.pdf</td>
<td>ConfigurationDeskUARTAPIReference.pdf</td>
</tr>
<tr>
<td>ConfigurationDesk Simulink Model Block Reference</td>
<td>SimulinkModelBlockRef.pdf</td>
<td>SimulinkModelBlockReference.pdf</td>
</tr>
</tbody>
</table>
Container Management

New Features of Container Management

Improved exchange of SWCs between SystemDesk 4.1 and TargetLink

Exchanging SWCs between SystemDesk 4.x and TargetLink has been improved by the following points:

Assigning AUTOSAR elements to the AUTOSAR files of an SWC container

With SystemDesk 4.1 you can assign AUTOSAR elements to AUTOSAR files of SWC containers directly. The following illustration shows SystemDesk's Container File Explorer that you can use for this.
The Container File Explorer shows a tree view of AUTOSAR files with their assigned AUTOSAR elements and a first item in the tree named Unassigned Elements that contains all the AUTOSAR elements that you have not assigned to a file.

An SWC container of an atomic SWC collects all the AUTOSAR files with dependent AUTOSAR elements.

**Improved assignment of AUTOSAR elements to AUTOSAR files**  To assign AUTOSAR elements, you can drag them to an AUTOSAR file. In addition, SystemDesk assists you with the following commands to assign AUTOSAR elements.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create automatic file assignments</td>
<td>To start assigning the AUTOSAR elements of an atomic SWC or SWC internal behavior to an AUTOSAR file.</td>
</tr>
<tr>
<td>Assign dependent AR elements</td>
<td>To list all the dependent AUTOSAR elements of an atomic SWC or SWC internal behavior and let you assign selected elements.</td>
</tr>
<tr>
<td>Expand file assignments</td>
<td>To show all the assigned elements of an atomic SWC.</td>
</tr>
</tbody>
</table>

**Multiple container import/export**  You can now import or export multiple SWC containers with one click. So you can import or export all the SWC containers of a SystemDesk project easily.

**Configuring the AUTOSAR version for exporting containers**

You can now configure the AUTOSAR version for exporting SWC containers in SystemDesk’s Preferences.

This allows you to configure the AUTOSAR export version in SystemDesk instead of editing the container workflow file (CTW).

**Related topics**

- Preferences ([SystemDesk 4.x Reference](#))
ControlDesk Next Generation

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>New Features of ControlDesk Next Generation (ControlDesk 5.1)</td>
</tr>
<tr>
<td></td>
<td>Migrating to ControlDesk Next Generation (ControlDesk 5.1)</td>
</tr>
</tbody>
</table>

Information in other sections

<table>
<thead>
<tr>
<th>ControlDesk Next Generation Migration Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explains migration from CalDesk/ControlDesk 3.x to ControlDesk Next Generation (ControlDesk 5.1).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ControlDesk Next Generation Migration of ControlDesk 3.x Automation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explains migration from ControlDesk 3.x automation to ControlDesk Next Generation automation.</td>
</tr>
</tbody>
</table>
### New Features of ControlDesk Next Generation
(ControlDesk 5.1)

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Project and Experiment Features (ControlDesk 5.1)</td>
<td>99</td>
</tr>
<tr>
<td>New Features of Platform Management and Platforms/Devices (ControlDesk 5.1)</td>
<td>99</td>
</tr>
<tr>
<td>New Variable Management Features (ControlDesk 5.1)</td>
<td>101</td>
</tr>
<tr>
<td>New Visualization and Instrument Features (ControlDesk 5.1)</td>
<td>102</td>
</tr>
<tr>
<td>New Measurement and Recording Features (ControlDesk 5.1)</td>
<td>105</td>
</tr>
<tr>
<td>New Bus Navigator Features (ControlDesk 5.1)</td>
<td>107</td>
</tr>
<tr>
<td>New Data Set Management Features (ControlDesk 5.1)</td>
<td>108</td>
</tr>
<tr>
<td>New ECU Diagnostics Features (ControlDesk 5.1)</td>
<td>108</td>
</tr>
<tr>
<td>New Signal Editor Features (ControlDesk 5.1)</td>
<td>110</td>
</tr>
<tr>
<td>New Failure Simulation Features (ControlDesk 5.1)</td>
<td>110</td>
</tr>
<tr>
<td>New Automation Features (ControlDesk 5.1)</td>
<td>111</td>
</tr>
<tr>
<td>Further Enhancements with ControlDesk Next Generation (ControlDesk 5.1)</td>
<td>112</td>
</tr>
</tbody>
</table>
New Project and Experiment Features (ControlDesk 5.1)

Display of the date and time when experiments were last opened and modified

ControlDesk’s Backstage view now displays the date and time when the displayed projects/experiments were last opened and modified.

New Features of Platform Management and Platforms/Devices (ControlDesk 5.1)

Information in this topic

General new features on page 100
Renaming platforms/devices on page 100
Updating the firmware of dSPACE systems on page 100
Support of Vector Informatik’s VN8900 interface on page 100
Specifying the period for variable observation on page 100
Loading an application via drag & drop on page 100
Optimized platform assignment on page 100
Video Capturing device: Replay of recorded videos on page 100
MicroAutoBox platform on page 101
Optimized platform assignment on page 101
Setting the system time on page 101
SCALEXIO Processing Unit platform: Optimized platform assignment on page 101
General new features

**Renaming platforms/devices**  ControlDesk now lets you rename platforms and devices. Any uses of the platform/device name in the experiment are automatically updated.

For details, refer to *Rename Platform/Device* (ControlDesk Next Generation Reference).

**Updating the firmware of dSPACE systems**  ControlDesk now provides the Update Firmware Wizard that lets you update, repair and downgrade the firmware of dSPACE systems.

For instructions, refer to *Updating and Repairing the Firmware of dSPACE Real-Time Hardware* (ControlDesk Next Generation Basic Practices Guide).

**Support of Vector Informatik’s VN8900 interface**  ControlDesk now supports Vector Informatik’s VN8900 system to access CAN, LIN, and FlexRay buses. However, the system is not supported by the ECU Diagnostics device.

**Specifying the period for variable observation**  Observing a variable means reading its value cyclically and displaying the instantaneous value of the observed variable in ControlDesk. ControlDesk now lets you specify the period for variable observation.

For details, refer to *Data Acquisition Page* (ControlDesk Next Generation Reference).

**Loading an application via drag & drop**  You can now load an application to a platform by dragging the corresponding application file from Windows Explorer to the platform in ControlDesk’s Platform/Device Manager.

**Optimized platform assignment**  When you select the Assign to first available platform assignment mode for a platform, you do not have to specify the platform’s connection type any more. ControlDesk assigns it to the first hardware it finds.

**Video Capturing device:**  
Replay of recorded videos  
The Video Capturing device now lets you replay recorded videos synchronously to a running measurement/recording. This lets you simulate a real camera during a measurement/recording.

For details, refer to *How to Configure Video Capturing Devices* (ControlDesk Next Generation Basic Practices Guide).
MicroAutoBox platform

**Optimized platform assignment** When you select the Assign to any equal assignment mode and the Bus connection type for a MicroAutoBox platform, you must specify the board name.

When you select the Assign to any equal assignment mode and the Net connection type for a MicroAutoBox platform, you must specify the net client as an IP address or alias. As an option, you can also specify the board name.

For details, refer to Assignment Properties (ControlDesk Next Generation Reference).

**Setting the system time** ControlDesk now lets you synchronize the system times of your MicroAutoBox and your host PC. This is useful if you want to use flight recording.

For details, refer to Set MicroAutoBox System Time (ControlDesk Next Generation Reference).

SCALEXIO Processing Unit platform: Optimized platform assignment

The SCALEXIO Processing Unit platform now supports the Assign to first available platform assignment mode, which lets ControlDesk automatically assign the platform arbitrarily to a SCALEXIO system that is registered but not yet assigned.

For details, refer to Assignment Properties (ControlDesk Next Generation Reference).

---

**New Variable Management Features (ControlDesk 5.1)**

**Replacing a variable description by a variable description with another name** ControlDesk now lets you replace the active variable description of a platform/device with a new one, even if the new variable description file has a name that is different from the initially imported one.

For example, you can then work with a CAN Bus Monitoring device in connection with different DBC files.

Refer to Replace Variable Description (ControlDesk Next Generation Reference).
New Visualization and Instrument Features (ControlDesk 5.1)

Information in this topic

- Importing and exporting variable connections on page 102
- User-defined tooltips for instruments on page 102
- Adjusting the instrument size to the size of the background image on page 102
- Getting the connection string of variables on page 102
- Adjusting the z order of instruments on page 103
- (WINDEX%) macro for measurement arrays and value blocks on page 103
- Custom default background color for layouts on page 103
- Copying instruments without variable connections on page 103
- Assigning platforms/devices to unresolved variable connections on page 103
- Steering Controller: Force feedback support on page 103
- Variable Array: Alternating row colors on page 104
- Static Text: User-defined text angle on page 104
- Radio Button: StateChanged event on page 104

Importing and exporting variable connections

ControlDesk now lets you import/export variable connections from/to the CONX file format. Refer to:

- **Import Connections** ([ControlDesk Next Generation Reference](#)).
- **Export Connections** ([ControlDesk Next Generation Reference](#))
  (exports the variable connections of the instruments on all currently opened layouts)
- **Export Active Layout Connections** ([ControlDesk Next Generation Reference](#))
  (exports the variable connections of the instruments on the currently active layout)

User-defined tooltips for instruments

You can now specify an instrument-specific tooltip, which is displayed when you move the mouse pointer over the instrument.

Refer to **Tooltip Property** ([ControlDesk Next Generation Reference](#)).

Adjusting the instrument size to the size of the background image

You can now adjust the size of instruments with enabled background image to the size of the image.

Refer to **Resize to Image** ([ControlDesk Next Generation Reference](#)).

Getting the connection string of variables

ControlDesk’s automation interface now lets you get the entire connection string of a variable connected to an instrument.
Adjusting the z order of instruments

You can now move an instrument selected on the layout layer-wise up and down.

Refer to Move Up (ControlDesk Next Generation Reference) and Move Down (ControlDesk Next Generation Reference).

(%INDEX%) macro for measurement arrays and value blocks

Instrument captions now support the (%INDEX%) macro. The macro lets you display the index position of a variable in a measurement array or value block.

Refer to Captions/Operating Elements Properties (ControlDesk Next Generation Reference).

Custom default background color for layouts

You can now specify a custom default color for the layout background.

Refer to Visualization Page (ControlDesk Next Generation Reference).

Copying instruments without variable connections

When you drag an instrument with the right mouse button, ControlDesk now lets you copy the instrument without its variable connections.

Refer to Copy Here Without Connections (ControlDesk Next Generation Reference).

Assigning platforms/devices to unresolved variable connections

ControlDesk lets you assign unresolved variable connections to platforms/devices in the experiment.

Refer to Assign Platforms (ControlDesk Next Generation Reference).

Steering Controller: Force feedback support

The Steering Controller now supports force feedback features of game controller devices.

For instructions, refer to How to Configure Force Feedback Effects (ControlDesk Next Generation Basic Practices Guide).
Variable Array: Alternating row colors

The Variable Array lets you set alternating row colors. The illustration below shows an example.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report All DTCs [19 02 FF] DTC_0x0001 4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Address DTCs [19 02 01] DTC_0x0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report All DTCs [19 02 FF] DTC_0x0001 4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Address DTCs [19 02 01] DTC_0x0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report All DTCs [19 02 FF] DTC_0x0002 4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Address DTCs [19 02 01] DTC_0x0001 4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report All DTCs [19 02 FF] DTC_0x0002 4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Address DTCs [19 02 01] DTC_0x0001 4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report All DTCs [19 02 FF] DTC_0x0003 4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Address DTCs [19 02 01] DTC_0x0002 4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report All DTCs [19 02 FF] DTC_0x0003 4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Address DTCs [19 02 01] DTC_0x0002 4s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Refer to Alternating Row Colors Properties (ControlDesk Next Generation Reference).

Static Text: User-defined text angle

You can now configure the angle of the text displayed in a Static Text instrument. The illustration below shows an example.

Refer to Static Text Properties (ControlDesk Next Generation Reference).

Radio Button: StateChanged event

ControlDesk’s automation interface now provides the StateChanged event for the Radio Button.
New Measurement and Recording Features (ControlDesk 5.1)

Using simulation time groups to improve synchronization accuracy

ControlDesk now provides *simulation time groups* to collect platforms/devices with synchronized simulation times. Using simulation time groups can improve synchronization accuracy.

Simulation time groups are useful in various synchronization scenarios. For example, the VEOS platform and/or the XCP on Ethernet devices representing the virtual ECUs of an offline simulation should be collected in a simulation time group for synchronized data acquisition, as shown in the illustration below:

The members of a simulation time group are handled as a whole during resynchronization, i.e., if periodic resynchronization is enabled, ControlDesk synchronizes the simulation time group as a whole, but not the individual platforms/devices belonging to the group.

For details, refer to *Using Simulation Time Groups* (ControlDesk Next Generation Basic Practices Guide).
Usability improvements to the Measurement Configuration controlbar

The usability of the trigger configuration area of the Measurement Configuration controlbar has been improved. As an example, the area now also displays the capture state of rasters (disabled, armed, running or stopped) as shown in the illustration below:

Refer to Measurement Configuration - Configuration Area (ControlDesk Next Generation Reference).
Importing and exporting recorders

ControlDesk now lets you import and export a recorder with its signal list and its property settings. Refer to:

- Import Recorder (ControlDesk Next Generation Reference)
  You can also import only the signal list to a recorder. Refer to Import Signals (ControlDesk Next Generation Reference).
- Export (Recorder) (ControlDesk Next Generation Reference)

New Bus Navigator Features (ControlDesk 5.1)

CAN base/update time displayed in the Bus Instrument (TX Type for CAN)

If the base/update feature is enabled for a message via the RTI CAN MultiMessage Blockset, the Bus Instrument (TX Type for CAN) displays the related variables.

Refer to Bus Instrument (TX Type for CAN) (ControlDesk Next Generation Reference).

RX source and destination addresses displayed in the Bus Instrument (RX Type for CAN) for J1939

For J1939, the Bus Instrument (RX Type for CAN) displays the source and destination addresses in addition to the ID of the RX message.

Refer to Bus Instrument (RX Type for CAN) (ControlDesk Next Generation Reference).

TX source and destination addresses displayed in the Bus Instrument (TX Type for CAN) for J1939

For J1939, if address-based addressing is configured via the RTI CAN MultiMessage Blockset, the Bus Instrument (TX Type for CAN) displays the source and destination addresses instead of the source and destination node index (J1939 SN and J1939 DN) for TX ID manipulation.

Refer to Bus Instrument (TX Type for CAN) (ControlDesk Next Generation Reference).
New Data Set Management Features (ControlDesk 5.1)

Combined filter in the Data Set Manager
The Data Set Manager now also supports the combined filter to filter a data set.
Refer to How to Filter a Data Set (ControlDesk Next Generation Basic Practices Guide) and Combined Filter (ControlDesk Next Generation Reference).

Making sub data sets read-only
You can now make sub data sets read-only, not just data sets only.
Refer to Make Read Only (ControlDesk Next Generation Reference).

Enhancement for reading/writing parameter values from/to the working data set
- With the Read from Working Data Set (ControlDesk Next Generation Reference) command, you can now also read the parameter values of a data set from the working data set, not only of a sub data set.
- With the Write to Working Data Set (ControlDesk Next Generation Reference) command, you can now also write the parameter values of a data set to the working data set, not only of a sub data set.

New ECU Diagnostics Features (ControlDesk 5.1)

The ECU Diagnostics v2.0.1 device (supporting ASAM MCD-3 D V2.0.1) was delivered for the last time with dSPACE Release 2013-A. 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.

For migration details and instructions, refer to Migrating to ControlDesk Next Generation (ControlDesk 5.1) on page 113.

Information in this topic
- Support of the ASAM MCD-2 D v2.2.0 ODX standard on page 109
- One variable description for each ODX database on page 109
- Creating an XML configuration for handling fault memory entries on page 109
- Display of control primitives with the NoOperation flag on page 109
- Enhanced ECU Diagnostics demo on page 109
### Support of the ASAM MCD-2 D v2.2.0 ODX standard

In addition to the ASAM MCD-2 D v2.0.1 ODX database version, the ECU Diagnostics device now also supports the ASAM MCD-2 D v2.2.0 ODX database version. ControlDesk can find the version automatically from the ODX database files.

Refer to *Diagnostics Management Page* (*ControlDesk Next Generation Reference*).

ControlDesk supports the mandatory functionalities of ASAM MCD-2 D v2.2.0 ODX. ControlDesk does not support all the functionalities of ASAM MCD-2 D v2.2.0 ODX.

### One variable description for each ODX database

ControlDesk now lets you generate one variable description individually from each ODX database. When you switch to another ODX database, the previously active variable description is deactivated since an ECU Diagnostics device can have at most one variable description active at a time.

For details, refer to *Basics of Measurement and Calibration via the ECU Diagnostics Device* (*ControlDesk Next Generation Basic Practices Guide*).

### Creating an XML configuration for handling fault memory entries

You can now make a fault memory service that is parameterized in the Diagnostics instrument available as an XML service configuration for use in an XML configuration file. You can copy the XML service configuration to the Clipboard or save it to a file.

For details, refer to *Create XML Configuration* (*ControlDesk Next Generation Reference*).

### Display of control primitives with the NoOperation flag

When the Diagnostics instrument’s `Show control primitives` property is selected, the instrument now also displays control primitives with the `NoOperation` flag.

For details, refer to *Tree View Properties* (*ControlDesk Next Generation Reference*).

### Enhanced ECU Diagnostics demo

The ECU Diagnostics demo now provides the diagnostic trouble codes layout to display and handle various DTC measurement variables.

For details, refer to *ECU Diagnostics Demo* (*ControlDesk Next Generation Basic Practices Guide*).
### New Signal Editor Features (ControlDesk 5.1)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exporting signals to the CSV file format</td>
<td>ControlDesk’s Signal Editor now lets you export signal descriptions and single signals to the CSV file format. This allows you to import signals to ControlDesk’s Measurement Data Pool, for example, or to process the signals in tools other than ControlDesk. The time axis resolution of a signal description set can be specified via the Resolution Property. Refer to Export (ControlDesk Next Generation Reference).</td>
</tr>
<tr>
<td>Display of time elapsed for a running signal generator</td>
<td>ControlDesk’s Signal Editor now displays the time that elapsed (in seconds) when a signal generator is running. Refer to Elapsed Time Property (ControlDesk Next Generation Reference).</td>
</tr>
</tbody>
</table>

### New Failure Simulation Features (ControlDesk 5.1)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| Switching multiple failures on SCALEXIO           | Switching multiple failures now supports:  
  - Short circuits of multiple channels to ground (KL31) – (short to GND)  
  - Short circuits of multiple channels to power switch channels with battery voltage (KL15, KL30) – (short to VBAT)  
  For details, refer to Failure Simulation Concept (SCALEXIO Hardware Installation and Configuration). |
### New Automation Features (ControlDesk 5.1)

<table>
<thead>
<tr>
<th>Information in this topic</th>
<th>Support of Python 2.7 on page 111</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Executing extension scripts when ControlDesk starts up on page 111</td>
</tr>
<tr>
<td></td>
<td>Creating a project when ControlDesk starts up on page 111</td>
</tr>
<tr>
<td></td>
<td>Getting the connection string of variables on page 112</td>
</tr>
<tr>
<td></td>
<td>Radio Button:StateChanged event on page 112</td>
</tr>
</tbody>
</table>

#### Support of Python 2.7

ControlDesk’s tool automation interface and the MCD-3 automation interface now use Python 2.7.

Up to dSPACE Release 2013-A, the Python 2.5 installation provided by dSPACE was a special setup including dSPACE-specific enhancements. This enhanced Python 2.5 installation was required for working with dSPACE software. With Python 2.7, the Python installation on the dSPACE DVD is equivalent to the standard Python distribution. dSPACE-specific components are now installed as add-ons. This allows you more flexibility when using Python. For example, you can add further Python packages without affecting the dSPACE installation.

For details on Python 2.7 and on migrating scripts from Python 2.5 to Python 2.7, refer to [Migrating Python Scripts from Python 2.5 to Python 2.7](#) on page 35.

#### Executing extension scripts when ControlDesk starts up

You can specify Python scripts as *extension scripts*. These scripts are executed automatically when ControlDesk starts up.

ControlDesk now lets you execute extension scripts. Extension scripts are Python scripts (PY or PYC files) that are executed each time ControlDesk starts up. Extension scripts can be executed for all users or user-specifically.

Refer to [Executing Extension Scripts When ControlDesk Starts Up](#) (ControlDesk Next Generation Advanced Practices Guide).

#### Creating a project when ControlDesk starts up

You can use ControlDesk events to create a project automatically when ControlDesk starts up.

For demonstration purposes, ControlDesk provides the CreateDefaultProjectOnStartUp demo. The demo uses the Started event to create a default project and experiment automatically when ControlDesk starts up. If there are registered platforms, these are added to the experiment. You can adapt the demo to meet your requirements.
Refer to Creating a Project When ControlDesk Starts Up (ControlDesk Next Generation Advanced Practices Guide).

<table>
<thead>
<tr>
<th>Getting the connection string of variables</th>
<th>ControlDesk’s automation interface now lets you get the entire connection string of a variable connected to an instrument.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio Button: StateChanged event</td>
<td>ControlDesk’s automation interface now provides the StateChanged event for the Radio Button.</td>
</tr>
</tbody>
</table>

**Further Enhancements with ControlDesk Next Generation (ControlDesk 5.1)**

| Document for the migration of ControlDesk 3.x automation | dSPACE HelpDesk now provides a document that explains migration from ControlDesk 3.x automation to ControlDesk Next Generation automation. To make this document easy to understand, it has the same structure and important contents as the ControlDesk 3.x Automation Reference, with additional migration information. Refer to the ControlDesk Next Generation Migration of ControlDesk 3.x Automation document. |
# Migrating to ControlDesk Next Generation (ControlDesk 5.1)

## Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Migrating to ControlDesk Next Generation (ControlDesk 5.1)</strong></td>
</tr>
<tr>
<td>113</td>
</tr>
</tbody>
</table>

## Information in other sections

<table>
<thead>
<tr>
<th>ControlDesk Next Generation Migration Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explains migration from CalDesk/ControlDesk 3.x to ControlDesk Next Generation (ControlDesk 5.1).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ControlDesk Next Generation Migration of ControlDesk 3.x Automation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explains migration from ControlDesk 3.x automation to ControlDesk Next Generation automation.</td>
</tr>
</tbody>
</table>

---

## Migrating to ControlDesk Next Generation (ControlDesk 5.1)

To migrate from ControlDesk 5.0 to ControlDesk 5.1 and reuse existing experiments, you may have to carry out the following migration steps.

## Information in this topic

- Discontinuation of the ECU Diagnostics v2.0.1 device on page 114
- Migration overview on page 114
- Migrating projects containing experiments with the ECU Diagnostics v2.0.1 device on page 114
- Migrating tool automation scripts with the ECU Diagnostics v2.0.1 device on page 117
- Migrating MCD-3 D automation scripts on page 119
- Tool automation changes on page 120
- Migrating scripts from Python 2.5 to Python 2.7 on page 120
- Changed default assignment mode for SCALEXIO on page 120
- Changed behavior of the TimingSequenceName property on page 120
- Changed behavior of the Selection Box's SelectedItemChanged event on page 121
- Migrating SYNECT's database on page 121
- Discontinuations after ControlDesk 5.1 on page 121
- User documentation renamed on page 122
- Migrating from CalDesk, ControlDesk 3.x, or ControlDesk Next Generation 4.3 or earlier on page 122
Discontinuation of the ECU Diagnostics v2.0.1 device

The ECU Diagnostics v2.0.1 device (supporting ASAM MCD-3 D V2.0.1) was delivered for the last time with dSPACE Release 2013-A. 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.

Migration overview

The main migration steps are:

- Adapt the COMPARAM-SPEC containing communication parameter (COMPARAM) definitions according to the D-PDU API (ISO 22900-2).

- If you use diagnostic jobs, recompile them against the job API of the MCD-3 D v2.0.2 standard.

- Adapt your automation scripts (refer to Migrating tool automation scripts with the ECU Diagnostics v2.0.1 device on page 117 and Migrating MCD-3 D automation scripts on page 119).

Migrating projects containing experiments with the ECU Diagnostics v2.0.1 device

To reuse a project containing experiments with the ECU Diagnostics v2.0.1 device, perform the following steps:

1. Open the project containing experiments with the ECU Diagnostics v2.0.1 device. The following dialog opens, asking you to convert the project and the contained experiments:
2. Click Yes to confirm.

*ControlDesk’s Project Converter* converts the project like this:

1. Each ECU Diagnostics v2.0.1 device in the project is converted into an ECU Diagnostics v2.0.2 device.

   Since the project cannot be loaded with ControlDesk 5.0 or earlier, a backup of the original project is stored in the project root folder during migration.

2. The *COMPARAM-SPEC* of the device’s ODX database is adapted using the diagnostic protocol-specific *COMPARAM-SPEC* template files in the `\Main\bin\Templates\ODX\COM_PARAMs` folder in the ControlDesk installation.

   Adaptation is necessary since the ECU Diagnostics v2.0.1 device and the ECU Diagnostics v2.0.2 device use different methods for *COMPARAM* identification:
   - The ECU Diagnostics v2.0.1 device used the optional *PDU-API-INDEX* parameter to identify a *COMPARAM*.
   - The ECU Diagnostics v2.0.2 device uses the short name of the parameter to identify a *COMPARAM*. ControlDesk supports all the standard communication parameters defined in D-PDU API according to ISO 22900-2.

   The *COMPARAMs* are adapted like this:
   - The values of *COMPARAMs* identified by a *PDU-API-INDEX* parameter are added to the *COMPARAM-SPEC* template file and adapted to the D-PDU API (ISO 22900-2) communication parameter format, e.g., timings are specified in µs instead of in ms.
   - *COMPARAMs* that are not identified by a *PDU-API-INDEX* parameter are adapted using a default value.
   - *COMPARAMs* that are identified by a *PDU-API-INDEX* parameter but for which there is no corresponding D-PDU API (ISO 22900-2) communication parameter are not adapted. A *LOG file* (refer to *Log File* ([ControlDesk Next Generation Reference])) entry is made.

3. After conversion, there are two different ways to proceed:
   - Using the ODX data converted by ControlDesk’s Project Converter:
     1. Check the *LOG file* (refer to *Log File* ([ControlDesk Next Generation Reference])) for possible conversion errors.
2. Check whether communication to the ECU can be established after device conversion. If no communication to the ECU can be established, you can also add ODX data to the converted device. See below (Using new ODX data).
   If still no communication to the ECU can be established, contact dSPACE Support.
3. If you use diagnostic jobs, recompile them against the job API of the MCD-3 D v2.0.2 standard.
4. Reload the recompiled diagnostic jobs on the Select ODX Files page.

   Do not reload all the ODX data since this overwrites the converted ODX data.

- OR: Using new ODX data, which is an alternative to the COMPARAM conversion described above:
  1. If you use diagnostic jobs, recompile them against the job API of the MCD-3 D v2.0.2 standard.
  2. Add ODX data to the converted device with COMPARAMS according to the D-PDU API (ISO 22900-2) communication parameter format.
There are some conversion limitations:

- Conversion is possible only for the following protocols:
  - KWP2000 on K-Line (ISO 14230)
  - Diagnostics on CAN (ISO 15765)
  - UDS on CAN (ISO 14229)
  - OBD

  Contact dSPACE Support for migration support if you use other diagnostic protocols.

- Do not reload the ODX data after device conversion since this overwrites the converted ODX data. Recompiled diagnostic jobs are an exception.

- If the ODX database contains no vehicle information specification, logical link names are automatically generated. The generation of logical link names, however, is different for the ECU Diagnostics v2.0.1 device and the ECU Diagnostics v2.0.2 device. Logical link names are used to name nodes in the generated variable description containing diagnostic variables. To avoid invalid connections between these variables and instruments after conversion, ControlDesk does not change the logical link names used for nodes in the generated variable description.

  As a consequence, the names of logical links used for nodes in the generated variable description are different from the logical link names appearing in the device configuration.

  For details on the naming of logical links, refer to Add ODX Database (ControlDesk Next Generation Reference).

**Migrating tool automation scripts with the ECU Diagnostics v2.0.1 device**

To reuse tool automation scripts with the ECU Diagnostics v2.0.1 device, perform the following steps:

1. Migrate the ControlDesk project containing experiments with the ECU Diagnostics v2.0.1 device (see Migrating projects containing experiments with the ECU Diagnostics v2.0.1 device on page 114).
2. In the script to be reused, change the following:

1. If your script contains a statement to add an ECU Diagnostics v2.0.1 device, you have to change the selected PlatformType:

<table>
<thead>
<tr>
<th>ECU Diagnostics v2.0.1 Device</th>
<th>ECU Diagnostics v2.0.2 Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding a device</td>
<td></td>
</tr>
<tr>
<td>PlatformType = Diagnostic = 11</td>
<td>PlatformType = Diagnostic2 = 27</td>
</tr>
</tbody>
</table>

   For details, refer to `PlatformType <<Enumeration>>` (ControlDesk Next Generation API Reference).

2. The automation interface for selecting the physical connection to be used for a logical link is different for the ECU Diagnostics v2.0.1 device and the ECU Diagnostics v2.0.2 device. You therefore have to adapt your script:

<table>
<thead>
<tr>
<th>ECU Diagnostics v2.0.1 Device</th>
<th>ECU Diagnostics v2.0.2 Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation interface</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td></td>
</tr>
<tr>
<td>IPmECUDiagnosticsLogicalLink</td>
<td>IPmECUDiagnostics2LogicalLink</td>
</tr>
<tr>
<td>• CANInterfaceSelection</td>
<td>• InterfaceSelection</td>
</tr>
<tr>
<td>• KlineInterfaceSelection</td>
<td></td>
</tr>
</tbody>
</table>

   For details, refer to `ECUDiagnostics2LogicalLink / IPmECUDiagnostics2LogicalLink <<Interface>>` (ControlDesk Next Generation API Reference).

3. If the ODX database contains no vehicle information specification, logical link names are automatically generated. The generation of logical link names, however, is different for the ECU Diagnostics v2.0.1 device and the ECU Diagnostics v2.0.2 device. You therefore have to adapt your script according to the new names.

   For details on the naming of logical links, refer to `Add ODX Database` (ControlDesk Next Generation Reference).
**Migrating MCD-3 D automation scripts**  
In connection with the discontinuation of the ECU Diagnostics v2.0.1 device, the DSystem and the D3System201 interfaces have also been discontinued:

- Via the DSystem interface, you accessed the ODX database directly without using a ControlDesk experiment.

To reuse MCD-3 D automation scripts that use the DSystem interface, perform the following steps:

1. Create a new ControlDesk project with an experiment and an ECU Diagnostics v2.0.2 device. Configure the experiment and the ECU Diagnostics v2.0.2 device according to your requirements.

2. In the MCD-3 D automation script, change the Dispatch command like this:
   
   From:
   ```
   Dispatch("ControlDeskNG.DSystem")
   ```
   
   To:
   ```
   Dispatch("ControlDeskNG.D3System202")
   ```

3. Since the DSystem.Configuration is no longer supported, you have to comment its methods out.

- Via the D3System201 interface, you used the configuration of an ODX database contained in a ControlDesk experiment.

To reuse MCD-3 D automation scripts that use the D3System201 interface, perform the following steps:

1. Migrate the ControlDesk project containing experiments with the ECU Diagnostics v2.0.1 device (see Migrating projects containing experiments with the ECU Diagnostics v2.0.1 device on page 114).

2. In the MCD-3 D automation script, change the Dispatch command like this:

   From:
   ```
   Dispatch("ControlDeskNG.D3System201")
   ```

   To:
Tool automation changes

**Migrating scripts from Python 2.5 to Python 2.7**  ControlDesk's tool automation interface and MCD-3 automation interface now use Python 2.7.

For information on script migration, refer to [Migrating Python Scripts from Python 2.5 to Python 2.7](#) on page 35.

**Changed default assignment mode for SCALEXIO**  Up to and including ControlDesk 5.0, Assign to any equal platform was the default mode for assigning registered SCALEXIO hardware to a platform in an experiment.

This means that the Assign to any equal platform mode is implicitly used if your automation script created for ControlDesk 5.0 (or before) contains code similar to the following lines:

```python
Platform = PlatformCollection.Add(22)  #SCALEXIO
Platform.Assignment.NetClient = '10.60.201.152'
Platform.Connect()
```

As of ControlDesk 5.1, the Assign to first available platform assignment mode is also supported for SCALEXIO, and it is the new default assignment mode.

As a consequence, if you still want to use the Assign to any equal platform assignment mode when you work with ControlDesk 5.1 or later, you have to modify the above code lines like this:

```python
Platform = PlatformCollection.Add(22)  #SCALEXIO
Platform.Assignment.Mode = 1  # Assign to any equal platform
Platform.Assignment.NetClient = '10.60.201.152'
Platform.Connect()
```

For reference information, refer to [AssignmentMode](#) <<Enumeration>> ([ControlDesk Next Generation API Reference](#)).

**Changed behavior of the TimingSequenceName property**  Up to and including ControlDesk 5.0, any string could be specified for the TimingSequenceName property. If the specified string was not part of the list of timing sequences, the running automation script was continued without activating a sequence.

As of ControlDesk 5.1, **ControlDesk outputs an error message** when a string is specified for the TimingSequenceName property for which there is no corresponding entry in the list of timing sequences.

For reference information, refer to [System / IFSSystem <<Interface>>](#) ([ControlDesk Next Generation API Reference](#)).
**Changed behavior of the Selection Box’s SelectedItemChanged event**

Up to and including ControlDesk 5.0, the Selection Box’s SelectedItemChanged event was not triggered in the following cases:

- You used the `SelectionBoxItem / IViSelectionBoxItem <<Interface>>`’s Remove method to remove an item from the Selection Box.
- You used the `SelectionBoxInstrument / IViSelectionBoxInstrument <<Interface>>`’s `ImportConfiguration` method to import a CSV file with Selection Box items.

As of ControlDesk 5.1, the Selection Box’s SelectedItemChanged event is also triggered in the cases described above.

For information on the event, refer to `SelectionBoxInstrumentEvents / IViSelectionBoxInstrumentEvents <<Events>>` (ControlDesk Next Generation API Reference).

**Migrating SYNECT’s database**

To use the data of previous client (ConfigurationDesk/ControlDesk Next Generation) versions with SYNECT server 1.2, you have to migrate SYNECT’s database.

SYNECT server provides a central database. Migrating SYNECT’s database therefore affects all of the client users that connect to SYNECT server. This means you have to migrate the database in a central process. For further information on migrating to SYNECT server 1.2, refer to Migrating to SYNECT Server 1.2 (SYNECT Server Document).

**Discontinuations after ControlDesk 5.1**

- The Flash User Identifier option, which lets the user specify a user ID for a dSPACE CAN interface (DCI-CAN1, Calibration hub) and save the ID to the flash memory of the interface, will be discontinued after ControlDesk 5.1.
- Elektrobit’s EB 61x0 (formerly BUSDOCTOR) interface will no longer be supported after ControlDesk 5.1.
To facilitate finding user documentation in the print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the online folder.

For ControlDesk Next Generation, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlDesk Next Generation API Reference</td>
<td>ControlDeskNGAPIReference.pdf</td>
<td>ControlDeskNextGenerationAPIReference.pdf</td>
</tr>
<tr>
<td>ControlDesk Next Generation ASAP3 Interface Reference</td>
<td>ControlDeskNGASAP3Reference.pdf</td>
<td>ControlDeskNextGenerationASAP3InterfaceReference.pdf</td>
</tr>
<tr>
<td>ControlDesk Next Generation Reference</td>
<td>ControlDeskNGReference.pdf</td>
<td>ControlDeskNextGenerationReference.pdf</td>
</tr>
</tbody>
</table>

Migrating from CalDesk, ControlDesk 3.x, or ControlDesk Next Generation 4.3 or earlier

To migrate from CalDesk, ControlDesk 3.x, or ControlDesk Next Generation 4.3 or earlier 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).
**DCI Configuration Tool**

**New Features of the DCI Configuration Tool 3.1**

<table>
<thead>
<tr>
<th>Start-up diagnostics and service implementation checks on DCI-GSI2s</th>
<th>When a DCI-GSI2 is put into operation with an ECU on a test bench for the first time, you should check that its cables, connectors, and circuit boards are correct. If the dSPACE Calibration and Bypassing Service is integrated in the ECU application, you should also check that the DCI-GSI2 configuration and the configuration of the dSPACE Calibration and Bypassing Service implemented in the ECU application are consistent. The DCI Configuration Tool now provides tests to perform start-up diagnostics and check the service implementation on a DCI-GSI2. For further information, refer to How to Perform Start-Up Diagnostics on a DCI-GSI2 (DCI Configuration).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support of DCI-GSI2s with Renesas RH850 microcontrollers</td>
<td>The DCI Configuration Tool now also supports device configuration for DCI-GSI2s that are equipped with a processor of the Renesas RH850 target processor family.</td>
</tr>
</tbody>
</table>
dSPACE ECU Flash Programming Tool

Migrating to dSPACE ECU Flash Programming Tool 2.2.4

User documentation renamed

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For the dSPACE ECU Flash Programming Tool, the following document is relevant:

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECU Flash Programming</td>
<td>ECUFlash.pdf</td>
<td>ECUFlashProgramming.pdf</td>
</tr>
</tbody>
</table>
dSPACE HIL API .NET

Migrating to dSPACE HIL API .NET 1.5

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For dSPACE HIL API .NET, the following document is relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
</table>
New Features of dSPACE FlexRay Configuration Package 3.2

FlexRay Configuration Tool

Support of AUTOSAR System Template 4.1.1
The FlexRay Configuration Tool now also supports the AUTOSAR System Template based on AUTOSAR Release 4.1.1 for describing FlexRay networks.

Improved handling in General Properties dialog
- You can now go through the single pages of the General Properties dialog and specify the relevant configuration settings in any sequence. The FlexRay Configuration Tool checks throughout the different pages if the configuration settings are valid. If there are inconsistencies, the FlexRay Configuration Tool provides information about the pages that contain errors.
When you configure your hardware on the Hardware page of the General Properties dialog, an update of the hardware configuration takes effect immediately, i.e., without confirmation via the OK button. As soon as you select a processor board, an I/O board or a controller module which is suitable to the rest of the current hardware configuration, it is set as the simulation hardware. If the selection is unsuitable, the FlexRay Configuration Tool informs you that further configuration settings are required.

Refer to General Properties (FlexRay Configuration Tool Reference).

<table>
<thead>
<tr>
<th>RTI FlexRay Configuration Blockset</th>
<th>Switchable check for RTI FlexRay Configuration blocks from custom libraries during update process</th>
<th>The RTIFLEXRAYCONFIG UPDATE block provides an option for you to enable/disable the check for RTI FlexRay Configuration blocks from custom libraries during the model update process. If the check is enabled, the entire Simulink model to be updated is scanned for RTI FlexRay Configuration blocks that are derived from custom libraries. If there are any blocks found, the update process aborts. If the check is disabled, the model update is performed, ignoring any RTI FlexRay Configuration blocks derived from custom libraries that are in your Simulink model. Refer to Model Update Page (RTIFLEXRAYCONFIG UPDATE) (RTI FlexRay Configuration Blockset Reference).</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlexRay Configuration Blockset</td>
<td>Switchable check for FlexRay Configuration blocks from custom libraries during update process</td>
<td>The FLEXRAYCONFIG UPDATE block provides an option for you to enable/disable the check for FlexRay Configuration blocks from custom libraries during the model update process. If the check is enabled, the entire Simulink model to be updated is scanned for FlexRay Configuration blocks that are derived from custom libraries. If there are any blocks found, the update process aborts. If the check is disabled, the model update is performed, ignoring any FlexRay Configuration blocks derived from custom libraries that are in your Simulink model. Refer to Model Update Page (FLEXRAYCONFIG UPDATE) (FlexRay Configuration Blockset Reference).</td>
</tr>
<tr>
<td>FlexRay Replay Script Generator</td>
<td>Support of Python 2.7</td>
<td>The FlexRay Replay Script Generator now supports Python 2.7.</td>
</tr>
</tbody>
</table>
Migrating to dSPACE FlexRay Configuration Package 3.2

User documentation renamed

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For the dSPACE FlexRay Configuration Package, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlexRay Configuration Tool Guide</td>
<td>FlexRayConfigToolGuide.pdf</td>
<td>FlexRayConfigurationToolGuide.pdf</td>
</tr>
<tr>
<td>FlexRay Configuration Tool Reference</td>
<td>FlexRayConfigToolRef.pdf</td>
<td>FlexRayConfigurationToolReference.pdf</td>
</tr>
<tr>
<td>FlexRay Configuration Features</td>
<td>FlexRayConfigFeatures.pdf</td>
<td>FlexRayConfigurationFeatures.pdf</td>
</tr>
<tr>
<td>RTI FlexRay Configuration Blockset Reference</td>
<td>FlexRayConfigRTI.pdf</td>
<td>RTIFlexRayConfigurationBlocksetReference.pdf</td>
</tr>
<tr>
<td>FlexRay Configuration Blockset Reference</td>
<td>FlexRayConfigDSRT.pdf</td>
<td>FlexRayConfigurationBlocksetReference.pdf</td>
</tr>
</tbody>
</table>
dSPACE Python Extensions

Where to go from here

Information in this section

| New Features of dSPACE Python Extensions 1.5 | 133 |
| Migrating to dSPACE Python Extensions 1.5 | 134 |

New Features of dSPACE Python Extensions 1.5

Support of Python 2.7

dSPACE Python Extensions now support Python 2.7. For further information and required migration steps, refer to Migrating Python Scripts from Python 2.5 to Python 2.7 on page 35.

dSPACE Platform Management API

With dSPACE Release 2013-B, the dSPACE Platform Management API has received a new identity. To create an instance of the platform management, use the following ProgID.

```python
import win32com.client
PlatformManagement = win32com.client.Dispatch("DSPlatformManagementAPI2")
...
```

For further information, refer to dSPACE Platform Management API Reference.
Discontinued software support

The following Python modules are not available with Python Extensions 1.5:
- rtpLib
- sdlLib
- matlabLib
- rs232Lib

New modules are available for rtpLib, matlabLib and rs232Lib. These are compatible with the old ones but provide additional features:
- rtpLib2
- matlabLib2
- rs232Lib2

For further information, refer to Test Automation Python Modules Reference.

The sdlLib Python module must be replaced either by the stimulus feature of the HIL API MAPort implementation or by Real-Time Testing scripts. Stimulus files in the new STZ format can be created by using the Signal Editor of ControlDesk Next Generation or by converting existing files via the sd2stz converter. For download and instructions, refer to http://www.dspace.com/go/sdm2stz.

For further information and required migration steps, refer to Migrating Python Scripts from Python 2.5 to Python 2.7 on page 35.

Migrating to dSPACE Python Extensions 1.5

Platform Management API

You have to change the program ID in applications to migrate to the Platform Management API coming with dSPACE Release 2013-B:

Replace DSPlatformManagementAPI by DSPlatformManagementAPI2.
To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For Python Extensions, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Automation Python Modules Reference</td>
<td>DSTAPythonRef.pdf</td>
<td>TestAutomationPythonModulesReference.pdf</td>
</tr>
<tr>
<td>Test Automation Python Modules Quick Reference</td>
<td>DSTAPythonRefQuick.pdf</td>
<td>TestAutomationPythonModulesQuickReference.pdf</td>
</tr>
</tbody>
</table>
# ECU Interface Manager

<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 ECU Interface Manager 1.4</strong></td>
<td>137</td>
</tr>
<tr>
<td><strong>Migrating to ECU Interface Manager 1.4</strong></td>
<td>140</td>
</tr>
</tbody>
</table>

## New Features of ECU Interface Manager 1.4

<table>
<thead>
<tr>
<th>Information in this topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support of Freescale MPC57xx microcontrollers on page 137</td>
</tr>
<tr>
<td>Enhancements for inserting service calls and execution controls on page 138</td>
</tr>
<tr>
<td>Display of the remaining memory resources on page 138</td>
</tr>
<tr>
<td>Option to dynamically define Memory statements on page 139</td>
</tr>
<tr>
<td>Macros for naming A2L/binary files on page 139</td>
</tr>
<tr>
<td>Grouping of functions in the Library pane on page 139</td>
</tr>
<tr>
<td>Support of the Intel HEX file format on page 140</td>
</tr>
</tbody>
</table>

### Support of Freescale MPC57xx microcontrollers

The ECU Interface Manager now supports Freescale MPC57xx microcontrollers.

The ECU Interface Manager also supports ECU image files using variable length encoding (VLE).
Enhancements for inserting service calls and execution controls

The Functions and Variables pane presents a structured list of all the functions and variables that can be prepared for bypassing. You can use this pane to configure the insertion of service calls and execution controls for functions and variables.

The illustration below shows the Functions and Variables Configuration pane as an example:

For details, refer to:
- Functions and Variables Pane ([ECU Interface Manager Reference])
- Functions and Variables Configuration Pane ([ECU Interface Manager Reference])

Display of the remaining memory resources

The new Memory Resources pane displays the remaining memory resources in the available memory segments. The displayed values of the remaining memory resources adapt automatically when you add service calls to or remove any from the application.

The illustration below shows the Memory Resources pane as an example:

For details, refer to Memory Resources Pane ([ECU Interface Manager Reference]).
Option to dynamically define Memory statements

You can now add definitions such as date and time of ECU application generation dynamically to Memory statements via predefined macros. The schema of software module description files has been extended for this purpose.

For details, refer to Schema of Software Module Description Files (ECU Interface Manager Guide).

Macros for naming A2L/binary files

You can configure the naming of A2L/binary export files via predefined macros.

The illustration below shows the available macros:

For details, refer to Exporting the Application (ECU Interface Manager Guide).

Grouping of functions in the Library pane

Functions can be grouped in the Library pane. For example, you can separate callable functions from top-level functions.

The illustration below shows an example:

For details, refer to Library Pane (ECU Interface Manager Reference).
Support of the Intel HEX file format

The ECU Interface Manager now also supports the Intel HEX file format for importing and exporting an ECU Image file.

For instructions, refer to How to Configure the Binary Exporter (ECU Interface Manager Guide).

Migrating to ECU Interface Manager 1.4

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

In ECU Interface Manager 1.4, 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 1.4, 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 1.4 and save it, the ECU Interface Manager prompts you to save it under a new name. This lets you keep a version of the project that you can work with in the former version of ECU Interface Manager.
Firmware Manager

Features of Firmware Manager 1.0

**New product**

The Firmware Manager is a new product by dSPACE. It allows you to update the firmware of dSPACE real-time hardware independently of other dSPACE software products.

It is available as a stand-alone setup, but is usually automatically installed with other dSPACE products that include platform management.

For a more flexible handling, the firmware itself is now also available with a separate installation. The several firmware components are packaged into platform-specific firmware archives. The Firmware Archives setup is automatically installed with dSPACE products that include platform management. If you only install the Firmware Manager, you also have to install the firmware archives.

**Main features**

With the Firmware Manager, you can update the firmware components of your real-time hardware to a later version. You can also repair the existing firmware version by reloading it to your hardware, for example, if there are problems with your hardware.

As an alternative to the user interface of the Firmware Manager, you can use the command line utility `FwManagerCmd`. This grants you a script-based access to the firmware management for repairing and updating. In addition, restoring a corrupt boot firmware is only possible via the command line utility.
### Update Firmware Wizard

The platform management of ControlDesk Next Generation and AutomationDesk now provide the **Update Firmware Wizard** that offers the same functionality as the Firmware Manager.

Only one firmware management tool is allowed to access a real-time hardware for firmware update at the same time.

### Supported hardware

The Firmware Manager is used to load firmware to the following dSPACE real-time hardware:

- DS1005 PPC Board
- DS1006 Processor Board

The Firmware Manager is replacing `firmware_update_ds1006.exe` and `prg1006.exe`.

- I/O boards with an updatable firmware in a modular system based on DS1005 or DS1006
- DS1103 PPC Controller Board
- DS1104 R&D Controller Board
- MicroAutoBox and MicroAutoBox II

### User documentation

For further information on the Firmware Manager and for general information on firmware management, refer to **Firmware Manager Document**.
Model Compare

New Features of Model Compare 2.4

Enhanced report generation options

Model Compare 2.4 provides you with extended options for report generation. The reports themselves have become more comprehensive. These are the improvements in detail (refer to Report Options Dialog (Model Compare Reference)):

Integration of model screenshots
You can integrate screenshots displaying model parts, for example, subsystems that contain highlighted or colorized blocks or lines. You can specify up to which subsystem level screenshots are provided.

Product use prohibited in United States
You are not licensed to use Model Compare in the United States. You are not allowed to use or permit others to use this product in the United States or in any way that violates the laws of the United States.
Improved content control You can apply a report content filter as follows:
- Show only differences
- Consider the filter settings as currently specified on the Display Page
- Consider report-specific filter settings

Improved PDF navigation Navigation in PDF reports is now more convenient:
- Bookmarks:
  The report generator bookmarks all subsystems and charts as well as all chapters up to level 2.

Traceability from model to Model Compare You can trace objects from Simulink back to Model Compare. This is helpful, for example, to identify blocks and lines that are highlighted or colorized in Simulink in Model Compare. For details, refer to How to Trace Models from MATLAB to Model Compare (Model Compare Guide).

Easy access to comparison settings via favorites You can switch comparison settings via toolbar, which is an alternative to the import/export of comparison settings. Using the toolbar you can switch between your favorite comparison settings and substitute the current ones. For details, refer to Basic Filter Settings (Model Compare Reference).

Migration to Model Compare 2.4

User preferences and comparison settings User preferences and comparison settings are stored separately from the tool. Thus, Model Compare 2.4 does not make use of the user preferences and comparison settings previously specified with Model Compare 2.0/2.1/2.2/2.3, but uses the defaults. However, user preferences and comparison settings that have been exported with Model Compare 2.0/2.1/2.2/2.3 can be imported by Model Compare 2.4.

Model Compare 2.4 lets you specify more user preferences and comparison settings than is possible with Model Compare 2.3 or older. For details refer to User Preferences Dialog (Model Compare Reference).
ModelDesk

New Features of ModelDesk 3.1

Platform management

ModelDesk now has a platform management component that is already used in other dSPACE tools. This allows you to manage the simulation platforms which are used to simulate the ASMs. The platform management provides the following features:

- Registering the simulation platform (real-time hardware or VEOS)
- Managing the registered simulation platforms
- Loading/reloading/unloading real-time applications or offline simulation applications
- Starting/stopping/pausing real-time applications or offline simulation applications
Synchronized platform management  You can access dSPACE real-time hardware and VEOS simultaneously with several dSPACE products such as ModelDesk, ControlDesk Next Generation, AutomationDesk, and ConfigurationDesk. The platform management instances in the different products are synchronized. Synchronization means that when you perform a platform management activity such as registering a platform in one instance, the contents of all the other currently running platform management instances are synchronized accordingly.

For details, refer to Synchronized Platform Management with Several dSPACE Products (ModelDesk Guide).

Parameterizing  
Table Editor  You can insert several rows or lines in a table in one step.

You can modify the values of vectors and tables by copying values via the Clipboard from Excel, MAT or ASCII files. If the number of copied values is greater than the dimension of the vector or table, its dimension is automatically increased.

Road Generator  
Scene synchronization with MotionDesk  The Road Generator provides different modes of synchronization with MotionDesk. You can specify which parts (road, scene or traffic) to synchronize.

Scene generation is also possible for MotionDesk PCs used in a multi-PC solution.

Improved zoom and navigation handling  The Road Generator has new controls for easier zooming and navigation in the preview panes.

Position of road and junction elements  It is now possible to move and connect junctions and road elements graphically via drag & drop in the Road Network 2-D Preview.

Position of surface conditions  It is now possible to move a specified surface condition graphically via drag & drop in the Road 1-D Preview and Junction 2-D Preview.
Migration to ModelDesk 3.1

User documentation renamed

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For ModelDesk, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModelDesk Guide</td>
<td>MdlDGuide.pdf</td>
<td>ModelDeskGuide.pdf</td>
</tr>
<tr>
<td>ModelDesk Reference</td>
<td>MdlDRef.pdf</td>
<td>ModelDeskReference.pdf</td>
</tr>
<tr>
<td>ModelDesk ASM Parameters Reference</td>
<td>MdlDASMParametersRef.pdf</td>
<td>ModelDeskASMParametersReference.pdf</td>
</tr>
<tr>
<td>ModelDesk Automation Reference</td>
<td>MdlDAutoRef.pdf</td>
<td>ModelDeskAutomationReference.pdf</td>
</tr>
</tbody>
</table>
MotionDesk

Where to go from here

Information in this section

New Features of MotionDesk 3.3 149
Migrating to MotionDesk 3.3 150

New Features of MotionDesk 3.3

Visualization

**Instruments**  You can now specify the font size of the instruments.

**Free observers**  The movement of the free observers is now smoother. A dead zone has been implemented to simplify the handling of the free observer.

**Full screen mode**  You can now use 1, 2 or 4 views in full screen mode.

Scene editing

**Multi selection**  You can select several 3-D objects and modify their properties at one time.

Scene generation

You can now use scene generation in a multi-PC solution. Synchronization from ModelDesk will update the road, scene and traffic on all MotionDesk PCs in a multi-PC scenario.

Library Manager

You can now group objects via drag & drop and copy & paste.

Frame rate

The frame rate in scenarios with quad-core computers has increased.
**Road generation**

Lane markings are now displayed on standard height maps (sinusoidal and trapezoidal).

**Graphics performance**

The graphics performance has been improved with various techniques, such as billboards and level of detail (LOD).

**Tool automation**

Tool automation has been expanded for the following areas: scene visualization, project and experiment management, and handling the data source for motion data.

---

**Migrating to MotionDesk 3.3**

**User documentation renamed**

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For MotionDesk, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MotionDesk Guide</td>
<td>MDGuide.pdf</td>
<td>MotionDeskGuide.pdf</td>
</tr>
<tr>
<td>MotionDesk Tutorial</td>
<td>MDTutorial.pdf</td>
<td>MotionDeskTutorial.pdf</td>
</tr>
<tr>
<td>MotionDesk Reference</td>
<td>MDRef.pdf</td>
<td>MotionDeskReference.pdf</td>
</tr>
<tr>
<td>MotionDesk Automation Reference</td>
<td>MDAutoRef.pdf</td>
<td>MotionDeskAutomationReference.pdf</td>
</tr>
<tr>
<td>MotionDesk Blockset Reference</td>
<td>MDBlockRef.pdf</td>
<td>MotionDeskBlocksetReference.pdf</td>
</tr>
</tbody>
</table>

**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 now uses a new 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).
Real-Time Testing

Migrating to Real-Time Testing 2.2

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For Real-Time Testing, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
</table>
RTI/RTI-MP and RTLib

### Where to go from here

Information in this section

| New Features of RTI/RTI-MP and RTLib | 153 |
| Migration Aspects of RTI/RTI-MP and RTLib | 156 |

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

**Support of MATLAB x64**

The support of 64-bit MATLAB versions now also includes the following RTI blocksets:

- RTI RapidPro Control Unit Blockset
- RTI Ethernet (UDP) Blockset
- RTI XCP on Ethernet Blockset
- RTI Watchdog Blockset (as a new RTI blockset, formerly it was included in the RTI 1401 Blockset)

For details, refer to *64-Bit Version of RCP and HIL Software* on page 19.

**Unsupported new features of MATLAB R2013b**

RTI and RTI-MP do not support the following new features in Simulink R2013b:

- **Comment Through**
  
  The Comment Through feature, to temporarily disable blocks and connect their input signals to output signals, is not supported for blocks of the RTI and RTI-MP blocksets.
- UTF-16 character support
  Block names, signal labels and annotations using the UTF-16 encoding are not supported by RTI and RTI-MP. Such characters might lead to problems when you generate code.

<table>
<thead>
<tr>
<th>Restricted access to blocksets when using MATLAB R2012a</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you use MATLAB R2012a, the RTI blocksets from dSPACE Release 2013-B are not included in the MATLAB Start menu. You can still access the blocksets from the Simulink Library Browser or by typing the library name in the MATLAB Command Window. MATLAB versions R2012b and later do not provide the Start menu.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enhancements to RTI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TRZ file generation</strong> If you have set the Enable real-time testing option in the Code Generation dialog, the generated MAP and TRC files are packed into a TRZ file.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enhancements to RTI-MP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New modeling mode</strong> Up to RTI-MP 7.0, you implemented one model that was automatically separated into submodels according to the configuration in the Multiprocessor Setup block when you started the build process, for example. The separation was repeated each time you started a build process, the task configuration, or only opened the submodel. As an interface between the submodels or CPUs, IPC and IPI blocks were automatically added to the submodels during separation. You can configure whether to create MP submodels for each CPU by separation or to include existing submodels. The blocks of an included submodel are not to be implemented in the MP main model. If each CPU is using the new modeling mode, the MP main model will only contain the Multiprocessor Setup block and the required IPC and IPI blocks. The key advantages of the new modeling mode are:</td>
</tr>
</tbody>
</table>
  - Submodels can be separately implemented
  - Submodels can also be used with RTI as single-processor applications
  - Separate task configuration for SP and MP models
  - A submodel can be used in several MP models
  - The MP model only contains the MP configuration
  Simulink simulation is not functional in this modeling mode. |
If you do not use the separation mechanism of RTI-MP to initially create the MP submodels, you have to configure the IPC and IPI blocks on your own.

For further information, refer to Details on MP Systems (RTI and RTI-MP Implementation Guide).

**Submodel-specific license check** The scope of the license check has been moved from the RTI-MP main model to its submodels. A valid license is required only for the RTI blockset currently used in a submodel.

<table>
<thead>
<tr>
<th>MicroAutoBox</th>
<th>MicroAutoBox has some hardware and software enhancements.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New DS1513 I/O Board</strong></td>
<td>The I/O features of the DS1513 I/O Board are:</td>
</tr>
<tr>
<td></td>
<td>■ A/D conversion (ADC Type 4 and AIO Type 1)</td>
</tr>
<tr>
<td></td>
<td>■ 32 parallel A/D converters, with 16 converters supporting burst conversion mode</td>
</tr>
<tr>
<td></td>
<td>■ D/A conversion (AIO Type 1)</td>
</tr>
<tr>
<td></td>
<td>■ 8 parallel D/A converters</td>
</tr>
<tr>
<td></td>
<td>■ Digital I/O Unit (DIO Type 4)</td>
</tr>
<tr>
<td></td>
<td>■ 24 digital input channels and 24 digital output channels that can be used for:</td>
</tr>
<tr>
<td></td>
<td>■ Bit I/O</td>
</tr>
<tr>
<td></td>
<td>■ PWM signal generation (PWM)</td>
</tr>
<tr>
<td></td>
<td>■ Square-wave signal generation (FREQ)</td>
</tr>
<tr>
<td></td>
<td>■ PWM signal measurement (PWM2D)</td>
</tr>
<tr>
<td></td>
<td>■ Frequency measurement (F2D)</td>
</tr>
<tr>
<td></td>
<td>■ Incremental encoder interface</td>
</tr>
<tr>
<td></td>
<td>■ CAN support (CAN Type 1)</td>
</tr>
<tr>
<td></td>
<td>■ 3 CAN modules, each with 2 CAN controllers and interfaces for RS232 and LIN/K-Line</td>
</tr>
<tr>
<td>The DS1513 I/O Board comes with the MicroAutoBox variants MicroAutoBox II 1401/1513 and MicroAutoBox II 1401/1512/1513.</td>
<td>For further information, refer to MicroAutoBox Features.</td>
</tr>
</tbody>
</table>
**MicroAutoBox configuration tool**  The MicroAutoBox configuration tool is now named DS1401ConfigGUI.exe and provides an additional dialog to enable the autoboot option for the FPGA Type 1 module. If autobooting is enabled, the FPGA application, which is stored in the board’s flash memory, is automatically loaded and executed when you switch on MicroAutoBox. This option can now be used instead of the fpga_tpi_enable_autoboot and fpga_tpi_disable_autoboot RTLib functions.

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

**Discontinued software support**

**Effects of the ControlDesk 3.x discontinuation**  The discontinuation of ControlDesk 3.x has some effects on RTI/RTI-MP and RTLib.

**Specifying the real-time hardware**  The platform management in ControlDesk Next Generation does not provide a working platform. As a consequence, the Platform selection setting in RTI no longer supports the Auto option. It is sufficient to enter the platform name that you specified in ControlDesk’s Platform Manager for platform registration. If you use a MicroAutoBox, you can optionally enter the board’s IP address instead of its name.

The load options pages for RTI and RTI-MP now provide only the Platform name setting to specify the destination for downloading the real-time application instead of the Platform selection and Processor board name settings. For MicroAutoBox, the page additionally provides the Connection identification by and the Network client setting.

Using RTLib, the parameters for the Down<xxxx> tool have changed. Internally, the ScoutCmd command line tool is replaced by the CmdLoader command line tool. For further information, refer to *Handling Applications via Command Line* ([ControlDesk Next Generation Basic Practices Guide](#)).
Generating stimulus signals  The stimulus signal generation feature via sdmlib and Test Automation Blockset has been discontinued. The Enable Test Automation Stimulus Engine setting was therefore removed from the RTI simulation options page. For test automation, use Real-Time Testing (available only for DS1005, DS1006 and MicroAutoBox).

For using RTLib, the ta_stimulus_engine_init, ta_stimulus_engine_service and ta_stimulus_engine_terminate functions have been discontinued.

Changed demo handling with ControlDesk Next Generation

Up to the current dSPACE Release, some RTI demo models contained a button to open a related ControlDesk 3.x experiment. With ControlDesk Next Generation, there are also demo experiments, but the direct access is not supported. If you open a demo model in Simulink, the model in SLX format and the related ControlDesk Next Generation project as a ZIP archive are automatically copied to the MATLAB working directory. To open the experiment, import the ZIP archive by using the Open Project + Experiment from Backup command in ControlDesk Next Generation. This creates a project in ControlDesk Next Generation’s working directory.

For RTLib, you find the ControlDesk Next Generation projects as ZIP archives in `<RCP&HIL_InstallationPath>/Demos`. They are named `<xxx>_hc.ZIP` to mark that they are relevant for handcoding. In the Source Files folder of the project, there is a `rebuildApplication.py` Python script to be executed in ControlDesk Next Generation. The script starts the processes for building the real-time application. This allows you to recompile the prepared real-time application.
In the ControlDesk Next Generation demo project, the platform is already specified. If the required platform is already registered when you open the demo project, the platform is automatically assigned to the connected hardware and you can run the demo project. If the platform is not automatically assigned, register the platform and select Configure Platform/Device from the platform's context menu to open the Assignments dialog.
Enter the platform settings and close the dialog by clicking Finish. The platform is then assigned.

For further instructions, refer to method 2 in How to Assign dSPACE Real-Time Hardware or VEOS to a Platform (ControlDesk Next Generation Basic Practices Guide).

Now you can start a measurement to run the demo project. When you start a measurement, the executable file with the real-time application is downloaded to the dSPACE real-time hardware and then started. For instructions, refer to How to Start Measuring (ControlDesk Next Generation Basic Practices Guide).

**Autoboot option not available**  ControlDesk Next Generation does not support the autoboot option for DS1103 and DS1005. Therefore, there is no Autoboot demo available in the current dSPACE Release.

For further information on migrating from ControlDesk 3.x to ControlDesk Next Generation, refer to ControlDesk Next Generation Migration Guide.

### MicroAutoBox watchdog blocks moved to RTI Watchdog blockset

The RTI blocks for watchdog handling have been moved from the RTI 1401 Blockset to a separate blockset. There is no manual migration necessary for existing models. For further information, refer to RTI Watchdog Blockset on page 179.
### MicroAutoBox configuration tool

The configuration tool for MicroAutoBox II is now installed on `<RCP&HIL_InstallationPath>/Exe` and renamed `DS1401ConfigGUI.exe`. If you have connected `ds1401_config.exe` to a User Function, for example, in ControlDesk Next Generation, you have to adapt the path and name.

### User documentation renamed

To facilitate finding user documentation in the `Print` folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the `Online` folder.

For the implementation software, the following documents are relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTI and RTI-MP Implementation Guide</td>
<td><code>ImplementRTI.pdf</code></td>
<td><code>RTIAndRTIMPImplementationGuide.pdf</code></td>
</tr>
<tr>
<td>RTI and RTI-MP Implementation Reference</td>
<td><code>RTIRef.pdf</code></td>
<td><code>RTIAndRTIMPImplementationReference.pdf</code></td>
</tr>
<tr>
<td>DS&lt;xxxx&gt; RTI Reference</td>
<td><code>DS&lt;xxxx&gt;RTI.pdf</code></td>
<td><code>DS&lt;xxxx&gt;RTIReference.pdf</code></td>
</tr>
<tr>
<td>DS&lt;xxxx&gt; RTLib Reference</td>
<td><code>DS&lt;xxxx&gt;RTLib.pdf</code></td>
<td><code>DS&lt;xxxx&gt;RTLibReference.pdf</code></td>
</tr>
<tr>
<td>DS&lt;xxxx&gt; Installation and Configuration</td>
<td><code>DS&lt;xxxx&gt;InstConfig.pdf</code></td>
<td><code>DS&lt;xxxx&gt;HardwareInstallationConfiguration.pdf</code></td>
</tr>
<tr>
<td>Modular Systems Hardware Reference</td>
<td><code>InstConfigRef.pdf</code></td>
<td><code>PHSBusSystemHardwareReference.pdf</code></td>
</tr>
<tr>
<td>RTI Bypass Blockset Reference</td>
<td><code>RTIBypassRTI.pdf</code></td>
<td><code>RTIBypassBlocksetReference.pdf</code></td>
</tr>
<tr>
<td>Document</td>
<td>Old Name</td>
<td>New Name</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>RTI Bypass Blockset MATLAB API Reference</td>
<td>RTIBypassMATLABAPI Reference.pdf</td>
<td>RTIBypassBlocksetMATLABAPI Reference.pdf</td>
</tr>
<tr>
<td>RTI Bypass Blockset Application Note</td>
<td>RTIBypassApplicationNote.pdf</td>
<td>RTIBypassBlocksetApplicationNote.pdf</td>
</tr>
<tr>
<td>RTI CAN Blockset Reference</td>
<td>RTICANRTI.pdf</td>
<td>RTICANBlocksetReference.pdf</td>
</tr>
<tr>
<td>RTI CAN MultiMessage Blockset Reference</td>
<td>RTICANMMRTI.pdf</td>
<td>RTICANMultiMessageBlockset Reference.pdf</td>
</tr>
<tr>
<td>RTI LIN MultiMessage Blockset Reference</td>
<td>RTILINMMRTI.pdf</td>
<td>RTILINMultiMessageBlockset Reference.pdf</td>
</tr>
<tr>
<td>RTI Ethernet UDP Blockset Reference</td>
<td>RTIEthernetUDPBlockset Ref.pdf</td>
<td>RTIEthernetUDPBlocksetReference.pdf</td>
</tr>
<tr>
<td>RTI FPGA Programming Blockset FPGA Interface Reference</td>
<td>RTIFPGAProgramming FPGARef.pdf</td>
<td>RTIFPGAProgrammingBlocksetFPGA InterfaceReference.pdf</td>
</tr>
<tr>
<td>RTI FPGA Programming Blockset FPGA Handcode Interface Reference</td>
<td>RTIFPGAProgramming FPGAHandcodeRef.pdf</td>
<td>RTIFPGAProgrammingBlocksetFPGA HandcodeInterfaceReference.pdf</td>
</tr>
</tbody>
</table>

1) Replace <xxxx> by the board number, for example 1005. For MicroAutoBox, all documents now start with MicroAutoBox instead of DS1401.
RTI Bypass Blockset

New Features of the RTI Bypass Blockset 3.1

<table>
<thead>
<tr>
<th>RTI Bypass Blockset</th>
<th>Support of further target processor families for internal bypassing</th>
<th>Migrating to RTI Bypass Blockset 3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The RTI Bypass Blockset now also supports internal bypassing for the Freescale MPC57xx and Infineon AURIX target processor families. dSPACE provides the new AML file dSPACE_INTERNAL_Bypass_v1_2_0.aml for the IF_DATA dSPACE_INTERNAL_BYPASS entry in this context. Refer to Interface Description Data for Internal Bypassing (Interface Description Data Reference).</td>
<td>163</td>
</tr>
<tr>
<td></td>
<td>Use of noncontiguous memory areas for bypass functions</td>
<td>166</td>
</tr>
<tr>
<td></td>
<td>Functions can be deployed in multiple, noncontiguous memory areas. The RTI Bypass Blockset now allows you to use multiple, noncontiguous memory areas of the same segment type (CODE, PARAMETER, VARIABLE) for internal bypassing. During code generation the segments are used in increasing order of the segment lengths, i.e., the smaller memory segments are filled first. Refer to Memory Page (RTIBYPASS_SETUP_BLx for INTERNAL) (RTI Bypass Blockset Reference).</td>
<td></td>
</tr>
</tbody>
</table>
The availability of this feature depends on the compiler used.

**Display of memory segment information in the MATLAB Command Window**  After the build process of the internal bypass ECU code is finished, a detailed summary of the current fill level (as absolute values in kb, and as relative values in %) of each memory segment is displayed in the MATLAB Command Window. It shows the used and remaining free CODE, PARAMETER and VARIABLE resources.

**Support for VLE code**  The RTI Bypass Blockset now supports variable length encoding (VLE) code for Power-PC (MPC5xxx) microcontroller architectures. You can configure the memory segments for use with VLE or BOOK-E instructions.

Refer to **Memory Page (RTIBYPASS_SETUP_BLx for INTERNAL)** ([RTI Bypass Blockset Reference]).

The availability of this feature depends on the compiler used.

**Support of Intel HEX file format for ECU applications**  The RTI Bypass Blockset supports the Intel HEX file format for ECU Image files. If you want to merge internal bypass code into an ECU application, you can now also import an ECU application of Intel HEX file format as the source, and/or store the new ECU application as an Intel HEX file.

You can mix the file formats for the input and output ECU applications. For example, you can use a HEX file as the original ECU application you want to merge the internal bypass code with, and then store the merged ECU application as an S19 file.

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

**Task call information for internal bypassing**  The RTI Bypass Blockset now also generates task call counter variables during the build process of an internal bypass application. The following variables are provided:

- Initialization: **RTIBypInitTaskCallCount** (generated only if service ID 0 is not used)
- **Root**: **RTIBypRootTaskCallCount**
- **Service ID**:  
  RTIBypServiceTaskCallCount_%<ServiceName>%<InstanceId> (where %<ServiceName> and %<InstanceId> describe the name and ID of the service instance that is used for triggering the subsystem)

A **CHARACTERISTIC** is generated into the A2L file for each variable. You can use these A2L file entries in a measurement and calibration tool (e.g., ControlDesk Next Generation) for debugging.
Generating A2L file entries by using MATLAB configuration structures  The RTI Bypass Blockset now lets you influence and extend the generation of A2L file objects (MEASUREMENTs, CHARACTERISTICs) by means of MATLAB workspace configuration structures. You can use additional variable property information to control the layout of Lookup Tables or function property information to define your own function group hierarchy, for example.

For further information, refer to Using a MATLAB Configuration Structure (RTI Bypass Blockset Reference).

Display of generated Simulink signals in Info block  The RTIBYPASS_INFO_BLx block lists the Simulink signals from the generated internal bypass code (available only in connection with bypass interfaces for internal bypassing). The name and data type of each signal are displayed.

Refer to Info Page (RTIBYPASS_INFO_BLx) (RTI Bypass Blockset Reference).

Support of enhancements to RTI Bypass Blockset  The RTI Bypass Blockset MATLAB API supports the enhancements to the RTI Bypass Blockset (see above). For example, it supports memory segment fragmentation for internal bypassing and VLE code generation.

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

Method for returning the available input ECU applications  The RTI Bypass Blockset MATLAB API comes with a new method for getting the paths and names of all ECU applications defined in the imported A2L files. This helps you select the original ECU application you want to merge internal bypass code with.

Refer to ECU Application Binary Files Selection (RTI Bypass Blockset MATLAB API Reference).
Migrating to RTI Bypass Blockset 3.1

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

The current release comes with RTI Bypass Blockset 3.1, which is compatible with earlier blockset versions 2.x. However, data management has changed in comparison to RTI Bypass Blockset 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 3.1, 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 3.1 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 3.1 (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.
RTI CAN MultiMessage Blockset

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>New Features of the RTI CAN MultiMessage Blockset 2.9</th>
<th>167</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrating to RTI CAN MultiMessage Blockset 2.9</td>
<td>169</td>
</tr>
</tbody>
</table>

New Features of the RTI CAN MultiMessage Blockset 2.9

New supported platform

The RTI CAN MultiMessage Blockset supports MicroAutoBox II with the new DS1513 IO board.

Support of AUTOSAR System Template 4.1.1

The RTI CAN MultiMessage Blockset now also supports the AUTOSAR System Template based on AUTOSAR Release 4.1.1 for describing CAN networks.

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

Manipulating gateway directions individually during run time

The RTI CAN MultiMessage Blockset now allows you to activate gatewaying separately for each gateway direction. You can enable/disable each gateway direction via TRC file and/or via inport during run time.

Refer to Setup Page (RTICANMM Gateway) (RTI CAN MultiMessage Blockset Reference).
Changes to J1939 communication

The RTI CAN MultiMessage Blockset provides a new network node addressing mode. In addition to addressing the source/destination nodes of a J1939 message by their node indices, you can now alternatively address them using their network addresses.

- The RTI CAN MultiMessage Blockset provides the new J1939 TX ID page, which allows you to change the priority and the source and destination nodes of J1939 TX messages during run time. On this page, you can also select whether to address the source/destination nodes of a J1939 message either by their network addresses or their node indices.
- J1939 messages are no longer displayed on the TX ID page.
- If you select to address J1939 messages by their network addresses on the J1939 TX ID page, the source and destination address information of these J1939 messages is made available as block inputs of the RTICANMM MainBlock. The RTI CAN MultiMessage Blockset provides the J1939SA and J1939DA model inputs to the TX Data inport of the RTICANMM MainBlock for this.

Refer to J1939 TX ID Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference).

Specifying individual cycle time tolerance ranges for several messages in one step

The RTI CAN MultiMessage Blockset now lets you set the tolerance ranges for the cycle times of several messages in a single step. The individual minimum and maximum time limit values are computed from the message’s default cycle time and the specified timeout tolerance percentage value each.

Refer to Cycle Time Error Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference).

Filtering signals of RX messages

If you open the Filter dialog from the Counter page or the Checksum Messages page, it now lists the signals of all RX messages in addition to the signals of the TX messages. For example, this allows you to easily select messages to be counted when they are received.

Refer to Filter Dialog (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference).

Avoiding algebraic loops between RTICANMM MainBlocks

The RTI CAN MultiMessage Blockset allows you to disable direct feedthrough for the TX input ports of the RTICANMM MainBlocks in your model in order to avoid algebraic loops.

Refer to Code Options Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference).
### RTI CAN MultiMessage Tutorial

The tutorial lesson that deals with working with a J1939-compliant DBC file has been extended and adapted to the changes in J1939 communication. It now also explains how to configure and observe the network management in ControlDesk Next Generation.

Refer to *RTI CAN MultiMessage Blockset Tutorial*.

### Migrating to RTI CAN MultiMessage Blockset 2.9

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

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

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

- In the MATLAB Command Window, enter `rtimmsu_update('System', gcs)`
  
  For further information on the command and its options, enter `help rtimmsu_update` in the MATLAB Command Window.

- Select the Create S-Function for all CAN Blocks command from the Options menu of the RTICANMM GeneralSetup block.

For further information, refer to *Limitations with RTICANMM* ([RTI CAN MultiMessage Blockset Reference](#)).
RTI LIN MultiMessage Blockset

Where to go from here

Information in this section

- New Features of the RTI LIN MultiMessage Blockset 2.2
- Migrating to RTI LIN MultiMessage Blockset 2.2

New Features of the RTI LIN MultiMessage Blockset 2.2

<table>
<thead>
<tr>
<th>New supported platform</th>
<th>The RTI LIN MultiMessage Blockset supports MicroAutoBox II with the new DS1513 I/O Board.</th>
</tr>
</thead>
</table>
| Support of AUTOSAR System Template 4.1.1 | The RTI LIN MultiMessage Blockset now also supports the AUTOSAR System Template based on AUTOSAR Release 4.1.1 for describing LIN networks.  
Refer to General Settings Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference). |
| Support of up to 63 LIN nodes | The RTI LIN MultiMessage Blockset now lets you simulate up to 63 LIN slave nodes in an RTILINMM configuration. In previous blockset versions, the maximum number of supported LIN slave nodes was 15.  
For MicroAutoBox/MicroAutoBox II, the maximum number of supported LIN slave nodes remains 15. |
Enhancements to the Master Selection page

The RTI LIN MultiMessage Blockset now provides further configuration settings on the Master Selection page to specify the LIN schedule behavior.

- You can specify whether the selected schedule table is breakable. A breakable schedule table means that a currently running schedule task can be interrupted by other schedules.
- The RTI LIN MultiMessage Blockset now provides direct header send functionality. If it is enabled, the LIN master is allowed to directly transmit a frame header from a simulation model without requiring a schedule. Two additional block inports are mapped to the TX Data inport of the RTILINMM MainSetup block for this.

Refer to Master Selection Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference).

Migrating to RTI LIN MultiMessage Blockset 2.2

Working with models from earlier RTI LIN MultiMessage Blockset versions

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

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

- In the MATLAB Command Window, enter
  \texttt{rtimmsu\_update('System', gcs)}
  For further information on the command and its options, enter \texttt{help rtimmsu\_update} in the MATLAB Command Window.
- Select the Create S-Function for all LIN Blocks command from the Options menu of the RTILINMM GeneralSetup block.

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

Where to go from here

Information in this section

New Features of the RTI FPGA Programming Blockset 2.6

Migrating to RTI FPGA Programming Blockset 2.6

New Features of the RTI FPGA Programming Blockset 2.6

Extended Xilinx® support

The RTI FPGA Programming Blockset now supports Version 14.6 (32-bit and 64-bit) 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>14.6 (32-bit version)</td>
<td>Windows XP Professional SP3 (32-bit version)</td>
<td>MATLAB R2012a</td>
</tr>
<tr>
<td></td>
<td>Windows 7 Business, Ultimate, and Enterprise SP1 (32-bit version and 64-bit version)</td>
<td>MATLAB R2012b</td>
</tr>
<tr>
<td></td>
<td>Windows 7 Business, Ultimate, and Enterprise SP1 (64-bit version)</td>
<td>MATLAB R2013a</td>
</tr>
<tr>
<td>14.6 (64-bit version)</td>
<td>Windows XP Professional SP3 (32-bit version)</td>
<td>MATLAB R2012b</td>
</tr>
<tr>
<td></td>
<td>Windows 7 Business, Ultimate, and Enterprise SP1 (32-bit version and 64-bit version)</td>
<td>MATLAB R2013a</td>
</tr>
<tr>
<td></td>
<td>Windows 7 Business, Ultimate, and Enterprise SP1 (64-bit version)</td>
<td>MATLAB R2013a</td>
</tr>
</tbody>
</table>

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

2) It is necessary to install a Xilinx System Generator 14.6 patch to get support for MATLAB R2013a, see http://www.xilinx.com/support/answers/56250.html.

3) Xilinx System Generator for DSP supports MATLAB R2013a only with Windows 7.
The DS5203 FPGA Board (LX50) and the DS2655 FPGA Base Module also support the WebPACK Editions of the Xilinx design tools.

### General enhancements

**Block display shows group number**

The blocks that are used for processor communication in the Processor Interface library (PROC\_XDATA\_READ\_BLx, PROC\_XDATA\_WRITE\_BLx, PROC\_INT\_BLx) and in the FPGA Interface library (FPGA\_XDATA\_READ\_BLx, FPGA\_XDATA\_WRITE\_BLx, FPGA\_INT\_BLx) contain the specified group number in their block displays if you have specified a register group.

Newly inserted blocks show the group number immediately, existing blocks show it after editing the block settings, reselecting the FPGA subsystem / FPGA model INI file in the PROC\_SETUP\_BL block, or generating the processor interface model.

**Variable buffer size**

In the FPGA\_XDATA\_READ\_BLx block and the FPGA\_XDATA\_WRITE\_BLx block, you can now specify the buffer size with a step width of 1 instead of 1024.

The FPGA still reserves a multiple of 1024 bytes for each specified buffer, but if you use an FPGA application in ConfigurationDesk, the data width of the FPGA function port and the corresponding model port is now the same.

### Enhancements to the DS2655 FPGA Base Module framework

The framework for the SCALEXIO DS2655 FPGA Module provides the following enhancement.

**New optional ports for APU Slave function**

The APU Slave function of the FPGA\_IO\_READ\_BLx block has new optional ports.

- **Phi Read HD**
  
  This port allows you to read the angle value for the APU bus in the range of 74 bit instead of the 16 bit provided by the Phi Read port.

- **Delta Phi**
  
  This port specifies the angle increment per 8 ns clock cycle.

- **Delta Phi Enable**
  
  If Delta Phi Enable is high, the value of Delta Phi has been added to Phi Read HD in the current clock cycle. The value of Delta Phi is also added to Phi Read, but the changes do not appear immediately because of the decreased resolution.

The data delivered by Phi Read HD, Delta Phi, and Delta Phi Enable allow you to calculate the angular speed and acceleration.

Each of the new ports has a corresponding simulation port.
Enhancements to the DS2655M1 I/O Module framework

The framework for the SCALEXIO DS2655M1 I/O Module provides the following enhancements:

- The Analog In function of the FPGA_IO_READ_BLx block has a new port. The Data New port outputs a flag that indicates the changes of the register status.
- The update rate of the A/D converters has been increased from 2 MSPS to 4 MSPS.

Enhancements to the DS5203 with onboard I/O framework

The frameworks for the DS5203 FPGA Board (LX50) and DS5203 FPGA Board (SX95) provide the following enhancement.

Inter-FPGA communication

The Inter-FPGA communication bus can be used to exchange data between two DS5203 FPGA Boards. The FPGA communication is realized directly between the connected FPGA applications without using buffers or registers on the processor bus. The 32 bit of the parallel bus can be configured using the related block dialogs.

The FPGA_IO_WRITE_BLx block provides the I-FPGA Master function to specify the FPGA application that transmits data (master) with its bus configuration.

The FPGA_IO_READ_BLx block provides the I-FPGA Slave function to specify the FPGA application that receives data (slave) with its bus configuration.

Related topics

Basics
- Migrating to RTI FPGA Programming Blockset 2.6 on page 176
Migrating to RTI FPGA Programming Blockset 2.6

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 2.6
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 2.6
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 2.6, 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 176.

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

Updating the FPGA framework using the script interface
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 subsystem in the processor model called MyProcModel. The specified values of the block parameters are not changed.
ProcModelHandle = get_param('MyProcModel','handle')
rtifpga_scriptinterface('FPGAFrameworkUpdate',
ProcModelHandle)

To update the FPGA framework and reset the values of the block parameters to their initial values
rtifpga_scriptinterface('FPGAFrameworkUpdate',
<SimulinkHandle>, 'ReInit')

The script handles all the subsystems in the model/subsystem which is specified by the Simulink handle. The parameters of the blocks are reset to their initial values after updating to the current framework version.
ProcModelHandle = get_param('MyProcModel','handle')
rtifpga_scriptinterface('FPGAFrameworkUpdate',
ProcModelHandle,'ReInit')

Custom functions incompatible with dSPACE Release 2013-B

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

A custom function generated by using RTI FPGA Programming Blockset 2.5 from dSPACE Release 2013-A and the real-time applications (*.RTA) containing the custom function are incompatible with dSPACE Release 2013-B. The FPGA model has to be rebuilt by using RTI FPGA Programming Blockset 2.6 from dSPACE Release 2013-B to get an applicable custom function.
RTI Watchdog Blockset

Where to go from here

Information in this section

Features of RTI Watchdog Blockset 1.0 179
Migrating to RTI Watchdog Blockset 1.0 180

Features of RTI Watchdog Blockset 1.0

New product availability

The RTI Watchdog Blockset is now available as a separate blockset. Before, it was integrated in the RTI 1401 Blockset. This makes the packaging of MicroAutoBox features more flexible.

Main features

The blockset allows you to implement a multistage run-time monitoring of tasks in your Simulink model and the built real-time application. The blockset also provides a demo model that shows you how to use the watchdog blocks.

The blockset supports MicroAutoBox II (DS1401-23 and later).

For further information, refer to RTI Watchdog Blockset Reference.
Migrating to RTI Watchdog Blockset 1.0

General migration aspects  If you have implemented a Simulink model using the watchdog blocks from the RTI 1401 Blockset available up to dSPACE Release 2013-A, and you migrate to dSPACE Release 2013-B, the DS1401_WD_<xxx> blocks are automatically migrated when you open the model. The block names remain, but the internal links are changed to the related WATCHDOG_<xxx> blocks.
SystemDesk 4.x

<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 SystemDesk 4.1</strong></td>
<td>182</td>
</tr>
<tr>
<td><strong>Migrating to SystemDesk 4.1</strong></td>
<td>190</td>
</tr>
</tbody>
</table>
New Features of SystemDesk 4.1

Where to go from here

Information in this section

| New General Features | 182 |
| Modeling Software Architectures | 182 |
| Modeling Systems | 184 |
| Using Validation Rule Sets | 185 |
| Improved Data Exchange Configuration | 186 |

New General Features

SystemDesk 4.1 has the following new general features.

AUTOSAR Releases supported by SystemDesk 4.1

SystemDesk 4.1 supports the modeling of software and system architectures according to AUTOSAR 4.1.1, 4.0.3, and 4.0.2.

Properties

SystemDesk supports all elements of the AUTOSAR schema. It provides specific dialog pages to specify the properties of important elements for specifying software architectures and systems more simply than with the complex AUTOSAR schema. However, you can view and edit all element properties on the Advanced page of the elements’ Properties dialogs.

With this version, SystemDesk provides the Properties control bar as an alternative to the Advanced page.

The Properties control bar allows you to quickly access the properties of a selected element and change them without opening a dialog.

Modeling Software Architectures

Improved software architecture modeling

With this version, modeling software architectures has been improved by points such as the following:

- Improved dialogs for easier access to the elements of software architectures
- Extended number of validation rules consistent modeling
- Improved support for connecting software components (see below)

**Improved support for connecting software components**

SystemDesk allows you to connect software components by assembly connectors, delegation connectors, and pass through connectors.

With this version of SystemDesk, connecting software components has been improved by the following points:

- SystemDesk analyzes connected port interfaces for compatible or incompatible interfaces. You can also validate connectors to check connections for compatibility.
- SystemDesk uses compatibility rules as defined by AUTOSAR. For additional information refer to the *Software Component Template* document at www.autosar.org.
- SystemDesk assists you when you connect a port by displaying all the ports with compatible interfaces so you can easily select valid destination ports in large-scale diagrams.

The illustration above shows the dialog that shows the ports that have compatible interfaces for a port. You can connect the port to one or more destination ports by selecting them in the dialog.
SystemDesk now assists you when you model systems.

A system is the combination of a software architecture and an underlying ECU network. The system also contains information on the ECUs’ network topology and mapping information, such as the mapping of software components to ECUs.

After you develop the software architecture, you need a system to distribute the software components of your software architecture to ECUs and to specify the ECUs’ network topology.

Modeling systems with the System Manager

To help you model systems, SystemDesk provides the System Manager shown in the illustration below.

The System Manager provides a clear view of systems. It allows you to create or select systems and add a software architecture, ECU instances, and network topology elements. Additionally, you can perform actions such as mapping software components to ECUs.
Steps when modeling systems

**Referencing a root SW composition** You have to reference a software architecture or root SW composition in a system. This root SW composition contains all the software components of the system to be modeled.

**Adding ECU instances** You have to add one or more ECU instances to a system. The ECU instances are used to define the ECUs used in the topology of the system.

**Mapping software components to ECUs** You have to map all the software component instances contained in the root SW composition to the ECU instances of the system.

**Selecting SWC implementations** You select an implementation for the SWC internal behavior of each atomic software component.

**Defining communication clusters** When using several ECUs, you have to define one or more communication clusters for your system. A communication cluster describes the topological connection of the communicating ECUs.

**Mapping SWC communication to network communication** Communication between components on different ECUs has to be mapped to system signals. This means mapping data elements to be transmitted between SWCs on different ECUs to system signals.

**Using Validation Rule Sets**

Using validation rules, you can check whether the AUTOSAR elements in your current project comply with specified criteria.

With this version of SystemDesk, you can select rule sets for validating AUTOSAR elements. You can then validate them according to a specific set of validation rules that have been designed for a specific purpose.
Exchanging data with behavior modeling and BSW generation tools

SystemDesk includes validation rule sets for exchanging data with TargetLink 3.4 and 3.5 and Elektrobit Trosos. You can also specify validation rule sets of your own to validate AUTOSAR elements according to your company rules.

The illustration above shows SystemDesk’s Preferences dialog that allows you to create or activate validation rule sets. You can also activate validation rule sets via SystemDesk’s menu.

Improved Data Exchange Configuration

SystemDesk allows you to assign AUTOSAR elements to files where they will be exchanged.

There are three use cases in which you can assign AUTOSAR elements. These are described below.

Storing AUTOSAR elements in content management and/or version management systems

You can use AUTOSAR master files to store AUTOSAR elements in content management and/or version management systems.

To write AUTOSAR elements to a master file, you have to assign them to exactly one AUTOSAR master file. However, you can leave them unassigned if you do not want to write them to a master file.
You can also read AUTOSAR master files to SystemDesk projects. AUTOSAR elements that are assigned to an AUTOSAR master file but are not present in the read-in AUTOSAR master file are deleted from the SystemDesk project. As a result, you can restore previous AUTOSAR master file versions.

The illustration above shows the Master File Explorer that allows you to assign AUTOSAR elements to AUTOSAR master files by dragging the elements to the files. The package structure of the AUTOSAR elements is also shown.
You can use AUTOSAR files to exchange AUTOSAR elements between project partners.

You can assign each AUTOSAR element to one or more AUTOSAR files as required. You can specify import/export options for each AUTOSAR file and import/export AUTOSAR files, groups of AUTOSAR files, or all the specified AUTOSAR files with one click. This way you can import/export AUTOSAR files of all the AUTOSAR releases supported by SystemDesk and also import only selected parts of a file.

The illustration above shows the AUTOSAR File Explorer and the AUTOSAR Import/Export File dialog for the CanBodyCluster.arxml file that allows you to specify the import/export options that are applied whenever you import/export the file.
Exchanging software components with TargetLink via the Container Manager

You have to use container files to exchange software components with TargetLink via the Container Manager.

Before exporting an SWC via a container, you must assign each AUTOSAR element that has a dependency to the SWC to exactly one container file. During container export, all the relevant container files are exported with the container.

The illustration above shows the Container File Explorer that allows you to assign AUTOSAR elements to container files. When you export SWCs such as the Controller software component, SystemDesk opens the Export Container dialog that shows the exported container files.
Migrating to SystemDesk 4.1

SystemDesk 4.1 automatically migrates SystemDesk 4.0 SDP project files upon loading.
## 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 TargetLink 3.5 and TargetLink Data Dictionary 3.5</td>
<td>192</td>
</tr>
<tr>
<td>Migrating to TargetLink 3.5 and TargetLink Data Dictionary 3.5</td>
<td>222</td>
</tr>
<tr>
<td>Changes in Future TargetLink Versions</td>
<td>243</td>
</tr>
</tbody>
</table>
New Features of TargetLink 3.5 and TargetLink Data Dictionary 3.5

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling in Simulink or Stateflow</td>
<td>192</td>
</tr>
<tr>
<td>Code Generation Core Functionality</td>
<td>195</td>
</tr>
<tr>
<td>Data Dictionary and Datamanagement</td>
<td>197</td>
</tr>
<tr>
<td>AUTOSAR</td>
<td>202</td>
</tr>
<tr>
<td>Testing Support</td>
<td>208</td>
</tr>
<tr>
<td>Code Generator Options</td>
<td>211</td>
</tr>
<tr>
<td>Toolchain Integration</td>
<td>213</td>
</tr>
<tr>
<td>Documentation</td>
<td>216</td>
</tr>
<tr>
<td>Other</td>
<td>217</td>
</tr>
<tr>
<td>API Commands</td>
<td>220</td>
</tr>
<tr>
<td>Hook Functions</td>
<td>221</td>
</tr>
</tbody>
</table>

Modeling in Simulink or Stateflow

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newly Supported Simulink Blocks</td>
<td>192</td>
</tr>
<tr>
<td>Enumeration Support</td>
<td>193</td>
</tr>
<tr>
<td>Improved System Preparation</td>
<td>194</td>
</tr>
<tr>
<td>Improved Stateflow Support</td>
<td>195</td>
</tr>
</tbody>
</table>

Newly Supported Simulink Blocks

<table>
<thead>
<tr>
<th>Variant Subsystem block</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetLink supports the Simulink Variant Subsystem block as follows:</td>
<td></td>
</tr>
<tr>
<td>◆ TargetLink always uses the active subsystem variant. Inactive subsystem variants are ignored.</td>
<td></td>
</tr>
</tbody>
</table>
Model Variant block

TargetLink supports the Simulink Model Variant block as follows, which reflects the usual model referencing behavior:
- TargetLink processes only the active model variant.
- The inactive model variants are not processed.

**Enumeration Support**

**Support of Simulink enumeration data types**

TargetLink now supports Simulink enumeration data types:
- TargetLink's system preparation replaces a Simulink Enumerated Constant block by a standard TargetLink Constant Block (TargetLink Block and Object Reference). When TargetLink data is cleared from a system, system preparation replaces the TargetLink Constant Block by a Simulink Enumerated Constant block. For details, refer to Basics on Applying Simulink Enumeration Data Types in TargetLink (TargetLink Preparation and Simulation Guide).
- By default, TargetLink maps Simulink enumeration data types used in Simulink blocks and Stateflow charts onto the new EnumImplementation TargetLink data type used for code generation during system preparation. You can change this data type mapping by using specific mapping functions, refer to System Preparation Hook Function Templates (TargetLink File Reference).
- TargetLink's production code does not contain enumeration values (e.g., Red, Yellow, and Green) but only the enumeration's integer values (e.g., 0, 1, and 2). Macros are also not supported. The same applies to logging and plotting because an enumeration's integer values are used. The base data type of the EnumImplementation data type is set to Int32 in the Data Dictionary templates, which you can change to other integer data types.
Improved System Preparation

Enhancements to Simulink ports

**Root system** You can decide whether the Simulink port blocks of the system that is going to be prepared are enhanced or not. These ports were always enhanced in older TargetLink versions. For details, refer to *System Preparation Dialog* (TargetLink Tool and Utility Reference).

**Underlying systems** You can decide whether the Simulink ports of nested subsystems are enhanced or not. TargetLink lets you specify the nested subsystems that are to be considered. Specification of the subsystems is possible via dialog and API. For details, refer to *Subsystems whose ports should be enhanced* (TargetLink Tool and Utility Reference).

TargetLink simulation frame

You can add and remove TargetLink simulation frames via API commands. These operations do not affect a subsystem’s ID. For details on these API commands, refer to `tl_addsimframe` (TargetLink API Reference) and `tl_removesimframe` (TargetLink API Reference).

Block replacement

TargetLink now allows you to override standard rules for the enhancement of Simulink blocks. For example, you can enhance a Simulink Gain block to a TargetLink Custom Code block instead of a TargetLink Gain block. Filter conditions as part of the libmap definitions allow you to exactly control in which situations the changed behavior is applied. For details on libmaps, refer to *How to Configure Block Replacement* (TargetLink Preparation and Simulation Guide).

Utility for block replacement

TargetLink provides a utility for easily enhancing unsupported Simulink blocks by TargetLink Custom Code blocks (type II). This is very helpful for Simulink blocks that are not natively supported by TargetLink. For details, refer to *Basics on Making Unsupported Simulink Blocks TargetLink-Compliant* (TargetLink Preparation and Simulation Guide).
**Improved Stateflow Support**

**Improved Stateflow support**

TargetLink now supports the **Execute (enter) Chart At Initialization** chart property for the code generation. As a result, you can specify whether a chart’s state configuration is initialized at model initialization time, which is new with TargetLink 3.5, or later, that is, at the occurrence of the first chart input event, which is the default setting.

**Enumeration data types**

TargetLink supports Simulink enumeration data types used in Stateflow, refer to *Enumeration Support* on page 193.

**Variant modeling with preprocessor #if**

The code for executing and exiting substates has been optimized by replacing the IF queries of the states’ activity variables by preprocessor IF statements. This optimization is applied to substates (N>=2) of a parent state with exclusive decomposition under specific conditions. For details, refer to *Basics on Substituting IF Queries by Preprocessor IF Statements* (TargetLink Preparation and Simulation Guide).

**Code Generation Core Functionality**

**Where to go from here**

Information in this section

- **MISRA-C:2004 compliance**
- **Improved Code Efficiency**

**MISRA-C:2004 compliance**

Several improvements were made to TargetLink’s Fixed-Point Library to comply with MISRA-C:2004. Improvements include:

- Bracketing of macro parameters
- Bracketing of bodies of loops and control flows
- Naming of module guards
- Using same parameter names in declarations and definitions

For the generated code TargetLink now brackets the arguments of the conditionlal operator (?) if the argument is an operation.
Improved Code Efficiency

Elimination of zero-initialized state variables

A state variable that is copied to another variable with static storage duration is replaced by the other variable if the initial values are equal. The implicit zero initialization value of uninitialized static storage duration variables now is taken into account and leads to elimination of more unnecessary state variables.

Related documentation

- UtilizeZeroInitializationOfStaticStorageDurationVariables
- Code Generator option
- Code Changes on page 232
Data Dictionary and Datamanagement

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhanced Usability of the Data Dictionary Manager</td>
<td>197</td>
</tr>
<tr>
<td>Further Improvements to the Data Dictionary</td>
<td>201</td>
</tr>
<tr>
<td>New DD MATLAB API Commands</td>
<td>201</td>
</tr>
</tbody>
</table>

Enhanced Usability of the Data Dictionary Manager

Automatic adaptation of references while moving or renaming DD objects

The Reference Handling Options let you specify the way references in DD objects will be handled when changing DD objects. By default, you can let the Data Dictionary Manager automatically adapt references for you. You can also disable this function via the Never Adapt References option. For information on configuring the Reference Handling Options, refer to Reference Handling Options (TargetLink Data Dictionary Manager Reference).

Accessing DD objects via DD handles

The Data Dictionary Manager now supports quick access to DD objects by entering Data Dictionary object handles (DD handles). Each DD object is identified by a unique handle number. Instead of using the object path, you can simply enter or paste the handle number into the DD object list. You can use this feature if you develop and debug M-scripts and need quick access to an object in the Data Dictionary Manager. However, DD handles are not persistent and expire if you delete their objects.

You can get the handle number in different ways, for example:
- using the GetDDAttributes DD MATLAB API command.
- using the About Object context command in the DD Navigator pane

For further information, refer to Basics on Accessing DD Objects via Paths and Handles (TargetLink Data Dictionary Basic Concepts Guide).
**Easier swapping of DD workspaces**

The new Make to Primary Workspace toolbar button allows you to easily swap DD workspaces. Select a DD workspace (except for DD0 itself) and make it the primary workspace (DD0) with one click.

For further information on swapping DD workspaces, refer to *How to Swap DD Workspaces* (TargetLink Data Dictionary Basic Concepts Guide).

**Improved handling of DDIncludeFile objects**

To easily transfer DD files, you can use the Point of Inclusion functions of the Data Dictionary Manager. Another quick method to manage points of inclusion is to handle their corresponding DDIncludeFile objects in the Config/DDIncludeFiles object tree:

- You can now access DDIncludeFile objects by double-clicking them in the Config/DDIncludeFiles subtree.
- A new dialog lets you edit the DDIncludeFile objects and load the corresponding included file, even if the point of inclusion object is yet to be defined.
In the context menu of the DDIncludeFile object, you can jump to the corresponding point of inclusion object (if already defined) or load its included file. In the parent DD tree, you can even load all included files.

For further information on DDIncludeFile objects and Points of Inclusion, refer to *Point of Inclusion* (TargetLink Data Dictionary Manager Reference).
Improved selection of multiple objects in panes

A new Select button makes it easy to multiselect objects in the following panes: Message Browser, Find Object Results pane, Find Reference pane and Custom Output pane. For example, if you have multiple search results in the Find Object Results pane, you can now multiselect them via left click and then click the Select button. All the selected search results are then automatically displayed/selected in the Data Dictionary Navigator and in the Object Explorer as shown in the illustration below.

For further information, refer to Overview of the User Interface (TargetLink Data Dictionary Basic Concepts Guide).

Related topics
Basics
- Overview of the User Interface (TargetLink Data Dictionary Basic Concepts Guide)

HowTos
- How to Swap DD Workspaces (TargetLink Data Dictionary Basic Concepts Guide)

References
- Point of Inclusion (TargetLink Data Dictionary Manager Reference)
- Reference Handling Options (TargetLink Data Dictionary Manager Reference)
Further Improvements to the Data Dictionary

Properties written to embedded objects

It is now possible to write properties directly to embedded objects via DD API commands.

Refer to the following example:

```
ddd('Set','/Pool/Variables/GroupInfo','CustomProp',1001);
```

New DD MATLAB API Commands

The following new DD MATLAB API commands are available:

- `HandleRefsToStringRefs`
- `SetUuids`

```
HandleRefsToStringRefs
```

```
ddd('HandleRefsToStringRefs',<objectIdentifier>,[<attributeName>,<attributeValue>])
```

Converts all handle reference properties of all objects in a subtree into string reference properties. Empty reference properties are left untouched. This also applies to reference properties with more than one variant of which at least one is empty.

```
SetUuids
```

```
ddd('SetUuids',<objectIdentifier>)
```

Sets the Uuid property for all objects in the subtree which have this property according to the Data Model, but do not have it set.

Related topics

References

- `dsddman` [TargetLink Data Dictionary MATLAB API Reference]
AUTOSAR

Information in this section

<table>
<thead>
<tr>
<th>Supported AUTOSAR Releases</th>
<th>202</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple Instantiation of Software Components</td>
<td>203</td>
</tr>
<tr>
<td>Per Instance Calibratable Parameters</td>
<td>204</td>
</tr>
<tr>
<td>Support of Code Generation Units Containing Only Operation Calls</td>
<td>205</td>
</tr>
<tr>
<td>Easier Specification of AUTOSAR Objects in the Data Dictionary Manager</td>
<td>205</td>
</tr>
<tr>
<td>Improved Restart Function Behavior for Incrementally Generated Runnable Code</td>
<td>206</td>
</tr>
<tr>
<td>Improved Container Management</td>
<td>207</td>
</tr>
</tbody>
</table>

Supported AUTOSAR Releases

TargetLink now supports AUTOSAR Release 4.1, Revision 4.1.1. The following AUTOSAR Releases are supported:

<table>
<thead>
<tr>
<th>AUTOSAR Release</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>4.1.1(^1)</td>
</tr>
<tr>
<td>4.0</td>
<td>4.0.3, 4.0.2</td>
</tr>
<tr>
<td>3.2</td>
<td>3.2.2, 3.2.1</td>
</tr>
<tr>
<td>3.1</td>
<td>3.1.5, 3.1.4, 3.1.2, 3.1.0</td>
</tr>
</tbody>
</table>
Specifying the 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).

Multiple Instantiation of Software Components

AUTOSAR supports multiple instantiation for code reuse. This lets you reduce memory overhead and testing efforts.

TargetLink helps you prepare SWC types for multiple instantiation. It performs all the necessary steps during code generation:

- Analyzes your model and DD-based specification and collects all the variables with static duration that belong to the SWC type prepared for multiple instantiation (states).
- Creates a PIM for each SWC type prepared for multiple instantiation. It automatically determines the correct data type of the PIM based on the collected states and moves them to the autogenerated PIM.
- Generates code such that all calls to instance-specific entities carry the instance handle as their first parameter. It ensures that the instance handle is propagated to subfunctions accordingly.
- Reports variables that could not be (automatically) implemented as PIM in its Code Generation Report.
You can simulate a single SWC instance in all the three simulation modes. Multiple instances can be simulated in the context of virtual validation.

**Related documentation**
- Preparing SWCs for Multiple Instantiation ([TargetLink AUTOSAR Modeling Guide](#))
- Analyzing the Code Generation Report ([TargetLink Preparation and Simulation Guide](#))
- AUTOSAR_POSCONTROL ([TargetLink Demo Models](#))

**Per Instance Calibratable Parameters**

TargetLink can create instance-specific calibration parameters (PerInstanceCalPrms) during code generation. The import- and export-of instance-specific calibration parameters from/to AUTOSAR files is also supported.

You have to explicitly specify how to implement calibration parameters via the **Class** property of the DD **Variable** objects used to specify the calibration parameters:

<table>
<thead>
<tr>
<th>Parameter Type</th>
<th>Variable Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared</td>
<td>AUTOSAR/SHARED_CALPRM</td>
</tr>
<tr>
<td>Instance specific</td>
<td>AUTOSAR/PER_INSTANCE_CALPRM</td>
</tr>
</tbody>
</table>

Reference the DDVariableGroup object containing your parameters at the CalibratablesRef/PerInstanceCalibratablesRef property of the RelatedVariables object belonging to the DD SoftwareComponent object. For instructions, refer to *How to Prepare Block Parameters of SWC Prototypes Derived From the Same SWC Type for Calibration* ([TargetLink AUTOSAR Modeling Guide](#)).
Support of Code Generation Units Containing Only Operation Calls

TargetLink lets you model operation calls as top-level subsystems into single code generation units (CGUs).

**Related documentation**  For details, refer to *Decomposing AUTOSAR Models* (TargetLink AUTOSAR Modeling Guide).

Easier Specification of AUTOSAR Objects in the Data Dictionary Manager

**Specifying AUTOSAR-compliant Variable objects**  
The new AUTOSAR context command in the DD Navigator pane allows you to specify Variable objects with AUTOSAR-compliant settings.

If you use the command on a VariableGroup object, a new AUTOSAR-compliant Variable object is created in the selected group.

If you use the command on an existing Variable object, the object is converted and given AUTOSAR-compliant settings.

The following AUTOSAR-compliant settings are supported for Variable objects:

- Explicit IRV
- Implicit IRV
- PIM
- SharedCalPRM
Composite SharedCalPrm

For further information, refer to Autosar (TargetLink Data Dictionary Manager Reference).

Specifying AUTOSAR-compliant Typedef objects

The AUTOSAR context command in the DD Navigator pane allows you to specify `typedef` objects with AUTOSAR-compliant settings.

If you use the command on a `typedefgroup` object, a new AUTOSAR-compliant `typedef` object is created in the selected group.

If you use the command on an existing `typedef` object, the object is converted and given AUTOSAR-compliant settings.

The following AUTOSAR-compliant setting is supported for `typedef` objects:

AUTOSAR Array Type

For further information, refer to Autosar (TargetLink Data Dictionary Manager Reference).

Improved Restart Function Behavior for Incrementally Generated Runnable Code

TargetLink's restart function behavior for incrementally generated runnable code has been improved:

For each runnable a restart function is created and called in TargetLink's restart runnable. This lets you initialize each runnable's state variables during restart in incremental code generation or several TargetLink subsystems.

The new DD `NoRestartCode` property specified at DD `Runnable` objects lets you suppress the restart function generation.
Error handling  If TargetLink detects necessary restart code suppressed via the NoRestartCode property, it displays an error message.

If you set the SuppressNoRunnableRestartCodeError Code Generator option to on, TargetLink displays a warning instead.

Related documentation
- TargetLink AUTOSAR Guidelines (TargetLink Modeling Guidelines)
- How to Model Runnables (TargetLink AUTOSAR Modeling Guide)
- Runnable

Improved Container Management

Container management was improved.

Related documentation
- New Features of Container Management on page 95
- Improved Data Exchange Configuration on page 186
Testing Support

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Injection/Tunneling</td>
<td>208</td>
</tr>
<tr>
<td>Third-Party Tool Support for Code Coverage Measurements</td>
<td>208</td>
</tr>
<tr>
<td>Enhanced Overflow Detection</td>
<td>209</td>
</tr>
<tr>
<td>Changes in the Target Simulation Modules</td>
<td>210</td>
</tr>
</tbody>
</table>

Signal Injection/Tunneling

Injecting/tunneling signals without a connection line

For testing purposes, TargetLink allows you to inject/tunnel signals without a connection line in all the three simulation modes. This lets you:

- Inject stimulus signals into TargetLink subsystems
- Tunnel signals from TargetLink subsystems to the outside

Both of the approaches listed above are modeled in Simulink.Signal objects and Data Store Read/Write blocks. A new demo model is provided for illustration.

Related documentation

- Injecting and Tunneling Signals During Simulation ([TargetLink Preparation and Simulation Guide](#))
- SIGNAL_INJECTION ([TargetLink Demo Models](#))

Third-Party Tool Support for Code Coverage Measurements

TargetLink now provides additional support for third-party code coverage measurement tools, such as the CTC Testwell Code Coverage (http://www.testwell.fi/ctcdesc.html).

Related documentation

- How to Measure Code Coverage with Third-Party Tools ([TargetLink Preparation and Simulation Guide](#))
- tlCodeCoverage
Enhanced Overflow Detection

Supported data types
TargetLink now supports overflow detection for single and integer Simulink/Stateflow data types. This is independent of TargetLink’s data type settings.

For details, refer to Overflow Detection (TargetLink Preparation and Simulation Guide).

Message types and saturation
TargetLink displays a message for each overflow it encounters during simulation. The message type depends on the overflows’s circumstances. Note the following distinction:

<table>
<thead>
<tr>
<th>Saturation for Code Generation Selected</th>
<th>Message Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>Note</td>
</tr>
<tr>
<td>➖</td>
<td>Warning</td>
</tr>
</tbody>
</table>

For details, refer to Details on the Visualization of Output Overflow (TargetLink Preparation and Simulation Guide).
### Changes in the Target Simulation Modules

The following table shows the compiler versions that are now supported by TargetLink 3.5. There is 1 new version and 1 discontinued version.

<table>
<thead>
<tr>
<th>Target</th>
<th>Compiler</th>
<th>New</th>
<th>No changes</th>
<th>Discontinued</th>
</tr>
</thead>
<tbody>
<tr>
<td>C16x</td>
<td>TASKING</td>
<td>—</td>
<td>8.6, 8.7</td>
<td>—</td>
</tr>
<tr>
<td>HCS12</td>
<td>Cosmic</td>
<td>—</td>
<td>4.7, 4.8</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Metrowerk</td>
<td>—</td>
<td>5.1, 3.1</td>
<td>—</td>
</tr>
<tr>
<td>M32R</td>
<td>Gaio</td>
<td>—</td>
<td>11, 9</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Renesas</td>
<td>—</td>
<td>5.1</td>
<td>—</td>
</tr>
<tr>
<td>MC56F83</td>
<td>Metrowerk</td>
<td>—</td>
<td>8.3</td>
<td>—</td>
</tr>
<tr>
<td>MPC55xx</td>
<td>Diab</td>
<td>—</td>
<td>5.9, 5.7</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>GreenHill</td>
<td>—</td>
<td>2012</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>GNU</td>
<td>—</td>
<td>4.1</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Metrowerk</td>
<td>—</td>
<td>2.8</td>
<td>—</td>
</tr>
<tr>
<td>MPC55xxVLE</td>
<td>Diab</td>
<td>—</td>
<td>5.9</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>GreenHill</td>
<td>—</td>
<td>2012</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Metrowerk</td>
<td>—</td>
<td>2.8</td>
<td>—</td>
</tr>
<tr>
<td>MPC560xVLE</td>
<td>Diab</td>
<td>—</td>
<td>5.9</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>GreenHill</td>
<td>—</td>
<td>2012, 5.2</td>
<td>—</td>
</tr>
<tr>
<td>MPC5xx</td>
<td>Diab</td>
<td>—</td>
<td>5.7</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>GreenHill</td>
<td>—</td>
<td>5.1</td>
<td>—</td>
</tr>
<tr>
<td>S12X</td>
<td>Cosmic</td>
<td>—</td>
<td>4.8</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Metrowerk</td>
<td>—</td>
<td>5.1</td>
<td>—</td>
</tr>
<tr>
<td>SH2</td>
<td>Renesas</td>
<td>—</td>
<td>9.3, 9.4</td>
<td>—</td>
</tr>
<tr>
<td>SH2A-FPU</td>
<td>Renesas</td>
<td>—</td>
<td>9.4</td>
<td>—</td>
</tr>
<tr>
<td>TriCore17xx</td>
<td>TASKING</td>
<td>4.2</td>
<td>3.2</td>
<td>4.0</td>
</tr>
<tr>
<td>TriCore1796</td>
<td>GNU</td>
<td>—</td>
<td>3.4</td>
<td>—</td>
</tr>
<tr>
<td>V850</td>
<td>GreenHill</td>
<td>—</td>
<td>2012</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>NEC</td>
<td>—</td>
<td>3.4</td>
<td>—</td>
</tr>
<tr>
<td>XC22xx</td>
<td>TASKING</td>
<td>—</td>
<td>3.0</td>
<td>—</td>
</tr>
</tbody>
</table>
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

#### Code Generator Options

The following new Code Generator Options are available with TargetLink 3.5.

<table>
<thead>
<tr>
<th>Description</th>
<th>Explanation</th>
<th>Behavior &lt;= TL3.4</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnableReportGeneration</td>
<td>Enables the generation of the Code Generation Report. If enabled, the Code Generator creates an XML-based report. The report contains information like the current value of the Code Generator options, the messages generated during code generation, and other information controlled by other options.</td>
<td>off</td>
<td>off</td>
</tr>
<tr>
<td>ImplementFloatEqualityViaMacros</td>
<td>Uses approximation for float equality. If an equality operation == or != is necessary, macro calls (C_EQFLT, C__EQDBL, C__NEFLT, C__NEBDL) are generated instead of a plain operation. These code patterns allow compliance with the MISRA standard. But the generated code may be less efficient, because the optimization will not eliminate such equality operations if macros are used.</td>
<td>off</td>
<td>off</td>
</tr>
<tr>
<td>ReportMismatchingPortBlockSignalSpecifications</td>
<td>Enables an optional report indicating mismatching port block signal specifications. If this option is enabled (and the option &quot;EnableReportGeneration&quot; is enabled) an optional report is generated, which indicates mismatches in the data type, scaling and min/max values between a port and its predecessor blocks. This may indicate an unwanted loss of accuracy.</td>
<td>off</td>
<td>on</td>
</tr>
<tr>
<td>ReportVariablesExcludedFromPIM</td>
<td>Enables an optional report indicating which variables could not be implemented as PIM and why. If this option is enabled (and the option &quot;EnableReportGeneration&quot; is enabled) an optional report is generated, which gives detailed information on which variables could not be implemented as PIM and why.</td>
<td>off</td>
<td>on</td>
</tr>
</tbody>
</table>
### SuppressNoRunnableRestartCodeError

<table>
<thead>
<tr>
<th>Description</th>
<th>Explanation</th>
<th>Behavior &lt;= TL3.4.</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuppressNoRunnableRestartCodeError</td>
<td>Emit a warning instead of an error if there is restart code for a NoRestartCode runnable. The generation of runnable-specific restart functions can be omitted by setting a runnable's NoRestartCode property to 'on'. If restart code is necessary for such a runnable, TargetLink will issue an error message. Setting this option changes this error to a warning, which makes it possible to analyze the generated code and to find the reason for the unwanted restart code.</td>
<td>on</td>
<td>off</td>
</tr>
</tbody>
</table>

### UtilizeZeroInitializationOfStaticStorageDurationVariables

<table>
<thead>
<tr>
<th>Description</th>
<th>Explanation</th>
<th>Behavior &lt;= TL3.4.</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>UtilizeZeroInitializationOfStaticStorageDurationVariables</td>
<td>Assume that &quot;no initial value all zeros&quot; for variables with static storage duration and utilize this for the elimination of intermediate variables in situations where initial values have to be equal. The C language demands that variables with static storage duration without an explicit initializer are initialized as if there was an initializer with 0 entry (or an initializer list with all 0 entries). Due to special compiler or linker configurations for the target platform, this may not hold for the target code. This option allows TargetLink to assume explicitly specified zero initialization. This affects in particular the elimination of output variables of conditionally executed subsystems. Often the default initial value 0 applies to these variables. Example: An enabled subsystem's output drives a TargetLink Outport block of the TargetLink subsystem. The output variable of the TargetLink OutPort is a global variable (i.e. has static storage duration), but has no explicit initial value derived from Simulink semantics. The output variable of the enabled subsystem has the initial value 0. This leads to the following situation: Int16 Output; void Subsystem(void) { static Int16 EnabledOut = 0; if (condition) { EnabledOut = ...; } Output = EnabledOut; } EnabledOut can be eliminated only if Output has the same initial value – otherwise the simulation behavior would change until the first time &quot;condition&quot; is true. If the option is activated, TargetLink performs this optimization for the above code. In addition, TargetLink makes the zero initial value explicit: Int16 Output = 0; void Subsystem(void) { if (condition) { ... Output = ...; }</td>
<td>off</td>
<td>on</td>
</tr>
</tbody>
</table>
For reference information on all Code Generator options, refer to Code Generator Options (TargetLink Block and Object Reference).

### Toolchain Integration

**Where to go from here**

Information in this section

- Handling Customization Files 213
- Enhanced Handling and Restoring of Demo Models 214
- TargetLink Simulation Module Extension Packages 215

### Handling Customization Files

**Customization file templates**

TargetLink’s customization files, such as hook functions and the files related to data type synchronization, are now shipped as template files (SAM). You can use the `tlCustomizationFiles('Create',...)` API function to derive functional customization files (M).

**Customization file folder**

You can place the derived customization files in a customization file folder outside the TargetLink installation folder. This lets you use customization files without administrator privileges and facilitates work in large workgroups.

**Related documents**

- Basics on Working Without Administrator Privileges (TargetLink Customization and Optimization Guide)
Enhanced Handling and Restoring of Demo Models

Selecting a root folder for the demo models

In TargetLink 3.4, the TargetLink demo model collection was saved in the TargetLink installation folder. This meant it was not possible to make any changes in the model without administrator privileges. When you first start a demo model in TargetLink 3.5, you are prompted to select a root folder. By default, the Documents folder is selected (<DocumentsFolder>:\dSPACE\TargetLink\<Version>\Demos).

Restoring the demo models to their original state

You can restore a demo model in the selected root folder with the `tl_demos [<DemoName>] -restore` command. If you use the command without `<DemoName>`, all demo models are restored to their original state.
**TargetLink Simulation Module Extension Packages**

TargetLink lets you install Simulation Module Extension Packages outside the installation directory.

You can install the extension packages via the Preferences Editor:

![Preferences Editor](image)

**Related documentation** For details, refer to *Customizing the TargetLink Environment* (TargetLink Customization and Optimization Guide).
Documentation

Improved Documentation

New user guides

With TargetLink 3.5, the structure of the user documentation has been improved. To facilitate access to the documentation, the topics are now grouped by their use case.

As a result, new use-case-specific guides are being introduced:

- **The all-new TargetLink Orientation and Overview Guide**
  This guide contains basic information for all people who work with TargetLink, from beginners up to expert users. It provides an introduction to TargetLink and points you to the user documentation that applies to your use case. It also lists TargetLink limitations and provides a glossary of terms used throughout the TargetLink documentation.

- **TargetLink Preparation and Simulation Guide**
  This guide shows you how to use TargetLink out-of-the-box for preparing Simulink models for code generation and for simulating and testing the generated production code.

- **TargetLink Customization and Optimization Guide**
  This guide shows you how to adapt TargetLink settings so that the generated production code meets your specific requirements.

- **TargetLink Interoperation and Exchange Guide**
  This guide shows you how to use TargetLink as part of a tool chain, for example, exchanging data with other tools or using TargetLink via command line.

Replaced guides

With the introduction of the new guides listed above, some of the previous guides in TargetLink have been replaced.

<table>
<thead>
<tr>
<th>Previous Guides</th>
<th>New Guides</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetLink Production Code Generation Guide</td>
<td>TargetLink Preparation and Simulation Guide</td>
</tr>
<tr>
<td>TargetLink Advanced Practices Guide</td>
<td>TargetLink Customization and Optimization Guide</td>
</tr>
<tr>
<td></td>
<td>TargetLink Interoperation and Exchange Guide</td>
</tr>
</tbody>
</table>
Other

Where to go from here

Information in this section

<table>
<thead>
<tr>
<th>Report to locate critical or inefficient code parts</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Code Generation Report</td>
<td>217</td>
</tr>
<tr>
<td>General Enhancements and Changes</td>
<td>219</td>
</tr>
</tbody>
</table>

New Code Generation Report

You can create a report during code generation. The report contains the following information:

- Messages that have been generated (also shown in the Message Browser)
- The code generator options used

Additional information helps you to locate critical or inefficient code parts during the development process:

- A list of mismatching port block signal specifications
- A list of variables not implemented as per instance memory (PIM)

The Code Generation Report is available in two file formats:

- An HTML file (CGReport.html) for viewing which you can open in the MATLAB Command Window after code generation
- An XML file (CGReport.xml) which you can use to further process the report contents in other tools/toolchains
The Code Generation Report contains useful links to the model or the Data Dictionary. The screenshot below is an example of a list containing links to mismatching port block signal specifications.

### Mismatching Port Block Signal Specifications

**Available subsystems**
- `poscontrol/controller/Subsystem/controller/Linearization`

**Linearization**

<table>
<thead>
<tr>
<th>min/max mismatch</th>
<th>Port block:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Simulink:</td>
</tr>
<tr>
<td></td>
<td><code>poscontrol/controller/Subsystem/controller/Linearization/POSITION</code></td>
</tr>
<tr>
<td></td>
<td>Signal indices: 1</td>
</tr>
</tbody>
</table>

**Predecessor block:**
- `poscontrol/controller/Subsystem/controller/POS`

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulink:</td>
<td><code>poscontrol/controller/Subsystem/controller/POS</code></td>
</tr>
<tr>
<td>Port Nr.:</td>
<td>1</td>
</tr>
<tr>
<td>Signal indices:</td>
<td>1</td>
</tr>
</tbody>
</table>


For information on how to create a Code Generation report, refer to [How to Create a Code Generation Report](TargetLink Preparation and Simulation Guide).

### Related topics

- **Basics**
  - [Basics on the Code Generation Report](TargetLink Preparation and Simulation Guide)

- **HowTo**
  - [How to Create a Code Generation Report](TargetLink Preparation and Simulation Guide)
General Enhancements and Changes

Latched InPorts

TargetLink now fully supports the Simulink Latch input for feedback signals of function-call subsystem outputs property of port blocks.

Initial values of TargetLink outports

You can specify initial values for the TargetLink OutPort Block and Bus Outport Block as in their Simulink counterparts. This is helpful (and only possible) if these blocks reside in conditionally executed subsystems.

Restriction

Bus signals must use the same scalar initial value.

Debugging scaling-invariant functions

By using the predefined Failure and FailureMsg variables in your M script, you can identify blocks that do not meet the preconditions for implementing scaling-invariant functions.

For an example, refer to Details on the Scaling Propagation Function (TargetLink Customization and Optimization Guide).

Adaptation of Simulink configuration parameters

TargetLink now notifies you of inconsistent bus signals. A certain Simulink configuration setting is expected. For details, refer to Basics on Preparing Simulink Systems for TargetLink Code Generation (TargetLink Preparation and Simulation Guide).

In a future version of TargetLink this configuration setting will become mandatory.

Floating-point parts encapsulated in Fixed-Point library

All the floating-point components of TargetLink’s Fixed-Point Library are now encapsulated by the TL_NO_FLOATS or TL_NO_FLOAT64 preprocessor directive.

This lets you control usage/removal of floating point code from the library during compilation:

<table>
<thead>
<tr>
<th>Directive</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL_NO_FLOATS</td>
<td>Suppress all the floating-point related macros and functions of the Fixed-Point Library (32-Bit and 64-Bit)</td>
</tr>
<tr>
<td>TL_NO_FLOAT64</td>
<td>Only suppress 64-bit floating-point related macros and functions of the Fixed-Point Library</td>
</tr>
</tbody>
</table>

Related documentation

- Basics on Conditional Code Compilation via Preprocessor Directives (TargetLink Customization and Optimization Guide)
Macros and Functions Provided by the Fixed-Point Library
(TargetLink File Reference)

Controlling Use of Fixed-Point Library Macros and Functions
(TargetLink File Reference)

Obsolete limitations
With TargetLink 3.5, several limitations have been removed. For details, refer to Obsolete Limitations on page 242.

API Commands

New API Commands

tlCustomizationFiles
TargetLink provides the tlCustomizationFiles('Create',...) API command that assists you in handling TargetLink's customization files and hook function templates.

Related documentation:
- tlCustomizationFiles (TargetLink API Reference)
- How to Create and Use TargetLink Customization Files (TargetLink Customization and Optimization Guide)
- How to Define TargetLink’s Search Path for Customization Files (TargetLink Customization and Optimization Guide)

tlCodeCoverage
TargetLink provides the tlCodeCoverage API command for support of the third-party code coverage tools.

Related documentation: For details, refer to How to Measure Code Coverage with Third-Party Tools (TargetLink Preparation and Simulation Guide).

tlFindDDReferences
TargetLink provides the tlFindDDReferences API command, which returns TargetLink blocks, Stateflow or Data Dictionary objects that reference a certain Data Dictionary object.

Related documentation: For details refer to tlFindDDReferences (TargetLink API Reference).
# Hook Functions

## New Hook Functions

| Comparing Code Generator options | There is now a new `tl_pre_compare_creator_options_hook.sam` hook function template. It allows you to customize the comparison of Code Generator options that is performed during the build process. For details, refer to `tl_pre_compare_creator_options_hook.sam` (TargetLink File Reference). |

Migrating to TargetLink 3.5 and TargetLink Data Dictionary 3.5

Upgrade process

To migrate from TargetLink 3.4 to TargetLink 3.5, 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 `tl_upgrade` API command.
For last-minute information on TargetLink 3.5 and potential problems, you are recommended to visit the TargetLink 3.5 website at http://www.dspace.com/goto?TargetLinkDocumentationUpdate.

- When upgrading models and libraries, first upgrade models or libraries that 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 231
- How to Make TargetLink User Libraries Upgrade-Capable (TargetLink Orientation and Overview Guide)
- tl_upgrade (TargetLink API Reference)

Projects that were with TargetLink 1.3 or even earlier versions cannot be upgraded directly to TargetLink 3.5 and TargetLink Data Dictionary 3.5. You must first perform an upgrade to a TargetLink 2.x version before you can upgrade to TargetLink 3.5 and TargetLink Data Dictionary 3.5.

- 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 236.
**Data Dictionary and Datamanagement**

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Dictionary and Datamanagement</strong></td>
<td>224</td>
</tr>
<tr>
<td>Code Changes</td>
<td>232</td>
</tr>
<tr>
<td>AUTOSAR</td>
<td>236</td>
</tr>
<tr>
<td>Documentation</td>
<td>238</td>
</tr>
<tr>
<td>Other</td>
<td>239</td>
</tr>
<tr>
<td>API Commands</td>
<td>240</td>
</tr>
<tr>
<td>Hook Functions</td>
<td>240</td>
</tr>
<tr>
<td>Obsolete</td>
<td>241</td>
</tr>
</tbody>
</table>

**Where to go from here**

**Information in this section**

| Migrating to TargetLink Data Dictionary 3.5 | 225 |
| How to Upgrade a Data Dictionary Without Included DD Files | 228 |
| How to Upgrade a Data Dictionary With Included DD Files | 229 |
| How to Upgrade Libraries and Models from TargetLink 2.x to 3.x | 231 |
# Migrating to TargetLink Data Dictionary 3.5

## Upgrading Data Dictionary files

To migrate from TargetLink 3.4 to 3.5, 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 3.5 provides an upgrade process that automatically upgrades DD files (DD0 workspace) to version 3.5.

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.

![Image of Data Dictionary Manager](image.png)

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')
anw =
    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 that 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 included dd files are used, there must be write access to all of them.
If included DD 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 included DD 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 Included DD Files

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

Method
To upgrade a Data Dictionary without included DD files

1. Open the model and the referenced TargetLink Data Dictionary, or type `dsdd('Open',<DDFile>)` in the MATLAB Command Window. The Data Dictionary needs upgrading dialog automatically opens if an older DD version is involved.

   ![Data Dictionary needs upgrading dialog](image)

2. Click Yes if no included DD files are used in the Data Dictionary. If included DD files are used, abort the upgrade process and refer to How to Upgrade a Data Dictionary With Included DD Files on page 229. The Delete generated objects dialog automatically opens if the Data Dictionary's Subsystem area contains objects generated with a previous TargetLink version.

   ![Delete generated objects dialog](image)

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 Included DD Files

Objective

If you open a TargetLink model with an old, non-upgraded Data Dictionary file, you have to upgrade the Data Dictionary file.

Method

To upgrade a Data Dictionary with Included DD files

1. Open the model and the referenced TargetLink Data Dictionary, or type `dsdd('Open',<DDFile>)` in the MATLAB Command Window. The Data Dictionary needs upgrading dialog automatically opens if an earlier DD version is involved.

![Data Dictionary needs upgrading dialog](image)

2. Select No in the upgrade dialog.

3. Under `/Config/DDIncludeFiles`, set the AutoLoad and AutoSave properties for each included DD file as shown below.

![Properties set](image)
This ensures that after the Data Dictionary and the included DD files have been upgraded, the upgraded included DD files are saved when the Data Dictionary is saved. You can set these properties for a large number of included DD files via the Object Explorer.

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

4 Start the DD upgrade (including the included DD 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 included DD 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 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 `dsdd_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 `dsdd('Upgrade', [DD_Identifier])` command, refer to Upgrade ([TargetLink Data Dictionary MATLAB API Reference](#)).

2. Type `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), e.g., `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.
## Code Changes

### Rounding of user-specified min/max values

TargetLink's rounding behavior for user-specified min/max values that cannot exactly be expressed in their corresponding scaling has changed with respect to range propagation.

That means that if a min or max value is specified for a block output/parameter or state and if that value cannot be exactly represented in the scaling of the output/parameter/state, a value rounded to the next representable value will be propagated to the connected block(s). This can result in:

- Additional saturation code
- Additional relational operations
- Changes in scale transformation operations (changing an operand with scaling A to scaling B)
- Casts to wider/different types compared to previous TargetLink versions

### The easiest and recommended way to avoid this issue is to always specify min/max values that can exactly be represented in the associated scaling.

### Background

In previous TargetLink versions, the floor() function was applied to the propagated value.

### Example

Consider the following TargetLink block output:

<table>
<thead>
<tr>
<th>LSB</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
<td>0.0</td>
</tr>
<tr>
<td>Min</td>
<td>-2.5</td>
</tr>
<tr>
<td>Max</td>
<td>2.4</td>
</tr>
</tbody>
</table>

The Min value can exactly be represented by a fixed-point value of -5 in the given scaling:

\[ \text{FxpValue} = (\text{FlpValue} - \text{Offset}) / \text{LSB} \Rightarrow -5 = (-2.5 - 0.0) / 0.5 \]

But to represent the Max value its fixed-point representation would have to be 4.8:

\[ \text{FxpValue} = (\text{FlpValue} - \text{Offset}) / \text{LSB} \Rightarrow 4.8 = (2.4 - 0.0) / 0.5 \]
Since the fixed-point value needs to be an integer, either 5.0 or 4.0 has to be chosen.

By applying the floor functionality, previous TargetLink versions chose 4.0, which corresponds to the floating point value of 2.0 (4.0 * 0.5).

TargetLink 3.5 performs rounding and the resulting floating-point value will be 2.5. Note that this rounding behavior might occur unexpectedly. Many floating-point values, such as 0.1, cannot exactly be represented using double. There is always a very small deviation and the double value is slightly smaller or greater than the expected value (0.1).

### Unary minus in Stateflow

For unary-minus operations specified in a Stateflow expression that contains at least one operand with a scaling (LSB != 1.0 and/or Offset != 0.0) and no floating-point operand, the order of scale transformation operations (shift, multiplication, division) and the unary-minus itself has changed.

### Optimization of runnables

Runnables that are not used in the code are no longer removed by TargetLink’s code optimization procedures.

### Discrete Time Integrator block

TargetLink now considers the full range of limits specified at Discrete-Time Integrator blocks, if they reference a DD VariableClass object with one of the following specifications:

- The `Info` property is set to a value other than `none` or `readonly`.
- The `Alias` property is set to `on`.
- The `Macro` property is set to `on` and the `Storage` property is set to `extern`.

This range is needed to define an adequate range and data type for the block’s state variable. So its range and data type might change in the generated code compared to code form previous TargetLink versions.
New Features and Migration November 2013

Discrete Filter, Discrete Transfer Function and Discrete State Space blocks

If the Keep vector structure checkbox (Coefficients page) of Discrete Filter/Discrete Transfer Fcn blocks or the Keep matrix structure checkbox (Matrices page) of Discrete State-Space blocks is not selected, TargetLink creates a structure for the coefficients/matrices. TargetLink now carries the min/max values of coefficient vectors/matrices specified in the blocks’s dialogs to the structure’s components. This can yield calculations in smaller bit-width.

Conditional operator (?:)

TargetLink now brackets the arguments of the conditional operator (?:), if the argument is an operation. This increases MISRA-C compliance.

Inlining of restart functions

TargetLink now inlines restart functions occurring within the same file.

Elimination of unnecessary state variables

TargetLink can eliminate variables used as states if they can be replaced by a variable with static storage duration and equal initial value. With TargetLink 3.5, this optimization is extended:

- TargetLink makes use of the ANSI C default initialization to zero of static storage duration variables without explicit initial value.

In order to document this assumption, the replacement variable gets an explicit initialization to zero; if this is not possible because there is no initialization method at the variable class, then no optimization takes place.
Block state variables now can be eliminated; the respective code situations can occur only due to code efficiency improvements in TargetLink 3.4.

**Initialization of static variables**  When the UtilizeZeroInitializationOfStaticStorageDurationVariables Code Generator Option is set to on (default), variables with static duration are initialized with 0. This especially affects the elimination of variables generated for the outputs of conditionally executed subsystems, whose values are written to global OutPort variables.

<table>
<thead>
<tr>
<th>TargetLink 3.5</th>
<th>≤ TargetLink 3.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int16 Out = 0; foo() { if (cond) { ... Out = ...; } }</td>
<td>Int16 Out; foo() { static Int16 X = 0; ... if (cond) { ... X = ...; } Out = X; }</td>
</tr>
</tbody>
</table>

**AUTOSAR operation calls** The production code can contain statements such as foo = Rte_Call_<Operation>() that were not contained in the code of prior versions.

**Explanation:** Operation calls can safely be removed by TargetLink’s optimization procedures, if the following conditions hold:

- Their return values are not used in the code.
- The operation call is known to have no states or side effects.

For operation calls explicitly modeled as server runnables in a Code Generation Unit, TargetLink can determine whether the operation call has states or side effect from the operation call’s implementation.

Operation calls whose implementation is opaque to TargetLink (not explicitly modeled as server runnables), can be removed by TargetLink’s optimization procedures only if the following conditions hold:

- Their return values are not used in the code.
- Their DD Operation object has its NoStatesOrSideEffects property set to on.
- Their DD NoDataFlowWithOtherOperations property is set to on, either directly or via the AssumeOperationCallsHaveNoUnknownDataFlow Code Generator option.
User-specified casts in Stateflow are treated differently. This can yield
differences in code in the following contexts:

- For casting MinMax/Abs calls, the conditional operator (?) is used.
- Compute through overflow behavior (unsigned casts) changed with
  respect to sequences of additions.
- Integer operations with floating-point constants:
  - If a suboperation contains a floating-point operand.
  - If this operand is part of a cast or shift operation.
- Index expressions of type float (additional integer cast).

Variables whose DD Optimization property is set to SCOPE_REDUCIBLE
but not ERASABLE are no longer reduced in scope, if they are not
read/written in functions or initializers.

Typically, these variables now have global scope instead of static global
scope.

AUTOSAR

AUTOSAR-Related Migration Aspects

TargetLink’s AUTOSAR synchronization functionality has been
improved and the synchronization scripts used in prior versions have
been removed. To ease access to the improved functions, they were
added to the relevant context menus in the /Pool/Autosar subtree.

The following table lists the new M-API commands and the
corresponding DD context commands:

<table>
<thead>
<tr>
<th>New API Command</th>
<th>New DD Context Command</th>
<th>Context Menu Of</th>
</tr>
</thead>
<tbody>
<tr>
<td>[bSuccess,processedObjectInfo] =...</td>
<td>Synchronize Application Data Type</td>
<td>/Pool/Autosar subtree</td>
</tr>
<tr>
<td>dsdd('SyncAutosar',&lt;objectIdentifier&gt;...</td>
<td>Settings</td>
<td></td>
</tr>
<tr>
<td>[,&lt;attributeName&gt;,&lt;attributeValue&gt;]);[1]</td>
<td>Synchronize Interface Settings</td>
<td>DD RequireCallPrmPort objects</td>
</tr>
<tr>
<td>[bSuccess,processedObjectInfo] =...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsdd('CreateVariableObjects',&lt;objectIdentifier&gt;...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[,&lt;bDeleteObsoleteInterfaceElements&gt;,&lt;bVerbose&gt;]);[2]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
New NVDATA synchronization  
TargetLink now provides a functionality to synchronize NVDATA settings:

<table>
<thead>
<tr>
<th>Command</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronize NvDataInterface Settings</td>
<td>Context menu of ports belonging to DD NvDataInterface</td>
<td>Synchronizes NvDataInterface settings.</td>
</tr>
</tbody>
</table>

1) Corresponds to the M-API command `[bSuccess, processedObjectInfo] = dadd('CreateVariableObjects', <objectIdentifier>...[,<bDeleteObsoleteInterfaceElements>,<bVerbose>]);`.

New properties in ImportExport AutosarOptionSet  
New properties were added to the `/Pool/Autosar/Config/ImportExport AutosarOptionSet`. These let you specify which synchronization commands are executed during imports of AUTOSAR data into the DD:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SynchronizeCalPrmInterfaceSettings</td>
<td>If on, synchronizes CalPrmInterface settings during import.</td>
</tr>
<tr>
<td>SynchronizeNvDataInterfaceSettings</td>
<td>If on, synchronizes NvDataInterface settings during import.</td>
</tr>
<tr>
<td>SynchronizeModeSettings</td>
<td>If on, synchronizes Mode settings during import.</td>
</tr>
<tr>
<td>SynchronizeApplicationDataTypeSettings</td>
<td>If on, synchronizes ApplicationDataType settings during import.</td>
</tr>
</tbody>
</table>

New AutosarOptionSet  
There is a new object called `/Pool/Autosar/Config/ContainerExchange DD AutosarOptionSet`. Its `PerformA2LExportOnContainerExport` property lets you specify to export an A2L file during container exchange.

You can copy the option set from a DD workspace based on the `dsdd_master_autosar.dd` template or create it via the `Create AutosarOptionSet` command from the context menu of the DD `/Pool/Autosar/Config` subtree in your existing DD files.

Deprecated hook function  
The `tl_post_arimport_hook_sync_calprm.m` hook function is deprecated and was removed from the TargetLink installation.
# Documentation

## Documentation Changes

User documentation renamed

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For TargetLink, the following documents are relevant:

<table>
<thead>
<tr>
<th>Title</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Guides</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TargetLink AUTOSAR Modeling Guide</td>
<td>TLAUTOSARGuide.pdf</td>
<td>TargetLinkAUTOSARModelingGuide.pdf</td>
</tr>
<tr>
<td>TargetLink Blockset Guide</td>
<td>TLRBlocksetGuide.pdf</td>
<td>TargetLinkBlocksetGuide.pdf</td>
</tr>
<tr>
<td>TargetLink Data Dictionary Basic Concepts</td>
<td>DDGuide.pdf</td>
<td>TargetLinkDataDictionaryBasicConceptsGuide.pdf</td>
</tr>
<tr>
<td><strong>References</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TargetLink Block and Object Reference</td>
<td>TLRBlockRef.pdf</td>
<td>TargetLinkBlockAndObjectReference.pdf</td>
</tr>
<tr>
<td>TargetLink Tool and Utility Reference</td>
<td>TLToolRef.pdf</td>
<td>TargetLinkToolAndUtilityReference.pdf</td>
</tr>
<tr>
<td>TargetLink API Reference</td>
<td>TLAPIRef.pdf</td>
<td>TargetLinkAPIReference.pdf</td>
</tr>
<tr>
<td>TargetLink File Reference</td>
<td>TLFileRef.pdf</td>
<td>TargetLinkFileReference.pdf</td>
</tr>
<tr>
<td>TargetLink Demo Models</td>
<td>TLDemoModels.pdf</td>
<td>TargetLinkDemoModels.pdf</td>
</tr>
<tr>
<td>TargetLink Evaluation Board Hardware</td>
<td>TLTargetRef.pdf</td>
<td>TargetLinkEvaluationBoardHardwareReference.pdf</td>
</tr>
<tr>
<td>TargetLink Data Dictionary Manager Reference</td>
<td>DDRef.pdf</td>
<td>TargetLinkDataDictionaryManagerReference.pdf</td>
</tr>
<tr>
<td><strong>Import Export Documents</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TargetLink Data Dictionary XML Import and Export</td>
<td>DDImportExport1.pdf</td>
<td>TargetLinkDataDictionaryXMLImportAndExport.pdf</td>
</tr>
</tbody>
</table>
Discontinued guides

The following guides have been discontinued:

- TargetLink Production Code Generation Guide
- TargetLink Advanced Practices Guide

They are substituted by new use-case oriented guides, refer to Improved Documentation on page 216.

Other

Various Migration Aspects

System preparation

The Replace blocks according to libmaps option is no longer available. As a result, if there are libmaps that define block replacements, these replacements are always performed. With TargetLink 3.5, you can no longer suppress them by clearing this option, which was only helpful for debugging libmaps. As a workaround, you can add a return at the beginning of a libmap, or remove the libmap file from the path, or rename it.

This option was used in the `tl_prepare_system` (TargetLink API Reference) API command and in the System Preparation Dialog (TargetLink Tool and Utility Reference).

Restoring demo models to their original state

You can restore a demo model in the selected root folder with the `tl_demos[<DemoName>] -restore` command. If you use the command without `<DemoName>`, all demo models are restored to their original state.

<table>
<thead>
<tr>
<th>Title</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetLink Data Dictionary ASAM MCD-2 MC Import and Export</td>
<td>DDIimportExport2.pdf</td>
<td>TargetLinkDataDictionaryA2LImportAndExport.pdf</td>
</tr>
<tr>
<td>TargetLink Data Dictionary OIL Import and Export</td>
<td>DDIimportExport3.pdf</td>
<td>TargetLinkDataDictionaryOILImportAndExport.pdf</td>
</tr>
<tr>
<td>TargetLink Data Dictionary AUTOSAR File Import and Export</td>
<td>DDIimportExport4.pdf</td>
<td>TargetLinkDataDictionaryAUTOSARFileImportAndExport.pdf</td>
</tr>
</tbody>
</table>

Title Old Name New Name
A2L export
A2L export now consists of two phases instead of three. For details, refer to Phases of the A2L File Export (TargetLink Data Dictionary: A2L Import and Export).

Changes in documentation generation
The mechanism of TargetLink's documentation generation has been changed. For details, refer to Generating Documentation on Model Characteristics (TargetLink Interoperation and Exchange Guide) and Maintaining and Documenting (TargetLink API Reference).

API Commands

Changes in TargetLink and TargetLink Data Dictionary API Functions

Deprecated commands

tl_find_system

This command is deprecated. It is only needed with MATLAB prior to Release R2011b. With newer MATLAB releases you can use the shipped Simulink find_system() command for TargetLink blocks as well.

Hook Functions

Hook Function Templates

Hook function templates

With TargetLink 3.5, hook function (and also other) templates are shipped as SAM files and no longer reside on the search path. For details on how to convert hook function templates to M files and place them on TargetLink's search path using the tlcCustomizationFiles('Create',...) API command, refer to the following documentation:

Related documentation

- How to Create and Use TargetLink Customization Files (TargetLink Customization and Optimization Guide)
**How to Define TargetLink’s Search Path for Customization Files**
(TargetLink Customization and Optimization Guide)

### Obsolete

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Discontinued Data Dictionary Features</td>
<td>241</td>
</tr>
<tr>
<td>Obsolete Limitations</td>
<td>242</td>
</tr>
</tbody>
</table>

### Discontinued Data Dictionary Features

**Discontinuation of EnableDefaultContainerExportHook property**

The DD EnableDefaultContainerExportHook property was removed from the /Pool/Autosar/Config/ImportExport subtree, due to a change in how hook function templates are shipped (refer to Hook function templates on page 240). TargetLink displays a warning if your DD file contains the discontinued property.

With TargetLink 3.5, AUTOSAR files and A2L files are gathered by the tl_export_container (TargetLink API Reference) API command. To enable/disable A2L export during container export, a new AUTOSAR option set is provided in the dsdd_master_autosar.dd template.

You can merge it from a new DD Workspace, based on the dsdd_master_autosar.dd template, or create it yourself.

<table>
<thead>
<tr>
<th>AUTOSAR Option Set</th>
<th>Custom Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Pool/Autosar/Config/ContainerExchange</td>
<td>PerformA2LExportOnContainerExport</td>
</tr>
</tbody>
</table>
Obsolete Limitations

With TargetLink 3.5, the following limitations existing in previous TargetLink versions have been removed. They are listed below.

### TargetLink simulations

<table>
<thead>
<tr>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simulation Rewind</strong></td>
<td>Rewinding (stepping backward) in simulations is not possible in TargetLink MIL/SIL/PIL simulation modes.</td>
</tr>
<tr>
<td><strong>Rewinding</strong></td>
<td>Rewinding is now possible in TargetLink MIL simulation mode. However, the limitation has not completely been solved, refer to <em>General Limitations</em> (<a href="#">TargetLink Blockset Guide</a>).</td>
</tr>
</tbody>
</table>

### Stateflow support

<table>
<thead>
<tr>
<th>Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Execute (enter) Chart at Initialization</strong></td>
<td>TargetLink does not support the Execute (enter) Chart At Initialization chart property. Any setting of this chart property is ignored and TargetLink generates a warning message.</td>
</tr>
<tr>
<td><strong>Execute (enter) Chart</strong></td>
<td>TargetLink now supports this chart property, refer to <em>Improved Stateflow Support</em> on page 195.</td>
</tr>
</tbody>
</table>
Changes in Future TargetLink Versions

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>To be Discontinued</td>
<td>243</td>
</tr>
<tr>
<td>Changes</td>
<td>243</td>
</tr>
</tbody>
</table>

To be Discontinued

- **A2L Import**: The A2L import will be removed in a future TargetLink version.
- **Generation of RTF documents**: The option to generate documentation in rich text format (RTF) may be removed in a future TargetLink version.

Changes

- **Clean distinction of vector and bus signals**: Recommended Simulink simulation configuration settings for clean distinction of vector and bus signals in the models will become mandatory in a future TargetLink version. This means:
  - That bus signals cannot be treated as vectors
  - That non-bus signals cannot be treated as bus signals
VEOS

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Features of VEOS 3.1</td>
</tr>
<tr>
<td>Migrating to VEOS 3.1</td>
</tr>
</tbody>
</table>

New Features of VEOS 3.1

<table>
<thead>
<tr>
<th>Support of MATLAB x64</th>
<th>VEOS now supports 64-bit MATLAB versions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>dSPACE Target for Offline Simulation: Real-Time Testing support always enabled</td>
<td>Real-Time Testing support now is always enabled when you generate an environment VPU. Using Real-Time Testing allows you to perform tasks such as stimulating VPU variables with ControlDesk Next Generation’s Signal Editor in an offline simulation, for example.</td>
</tr>
</tbody>
</table>
VEOS Player improvements

**Updating VPUs**  When you import one or more VPUs that are already in the currently open simulation system, the VEOS Player lets you update the VPUs while keeping existing VPU port connections if possible.

For details, refer to Import Vpu (VEOS Player Document).

**Improved user interface**  The user interface of VEOS Player has been improved in VEOS 3.1. For example, you can:

- Hide connected incoming and/or outgoing ports
- Turn the automatic export of variable description files on/off when the simulation system is saved

For details, refer to User Interface of the VEOS Player (VEOS Player Document).
Migrating to VEOS 3.1

User documentation renamed

To facilitate finding user documentation in the Print folder of dSPACE HelpDesk, the PDF file names have been changed to match the titles of the documents. The modifications are also applied to the related CHM files in the Online folder.

For VEOS, the following document is relevant.

<table>
<thead>
<tr>
<th>Document</th>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>dSPACE Target for Offline Simulation Document</td>
<td>DSOOfflineSimulationTarget.pdf</td>
<td>dSPACETargetForOfflineSimulationGeneration.pdf</td>
</tr>
</tbody>
</table>
Compatibility Information

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported MATLAB Releases</td>
<td>250</td>
</tr>
<tr>
<td>Operating System</td>
<td>251</td>
</tr>
<tr>
<td>Run-Time Compatibility of dSPACE Software</td>
<td>253</td>
</tr>
<tr>
<td>Limitations for 64-Bit Windows Operating Systems in Combination with 32-Bit dSPACE Software</td>
<td>254</td>
</tr>
<tr>
<td>Limitations for Products on 64-bit dSPACE DVD</td>
<td>254</td>
</tr>
<tr>
<td>Limitations for Windows 7</td>
<td>257</td>
</tr>
</tbody>
</table>
## Supported MATLAB Releases

<table>
<thead>
<tr>
<th>MATLAB Release...</th>
<th>...Is Supported by dSPACE Release 2013-B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RCP and HIL Software</td>
</tr>
<tr>
<td></td>
<td>RCP and HIL Software from 32-bit DVD</td>
</tr>
<tr>
<td></td>
<td>with 32-bit MATLAB versions</td>
</tr>
<tr>
<td>R2013b</td>
<td>✓ 1)</td>
</tr>
<tr>
<td>R2013a</td>
<td>✓ 2)</td>
</tr>
<tr>
<td>R2012b</td>
<td>✓ 3)</td>
</tr>
<tr>
<td>R2012a</td>
<td>✓ 4)</td>
</tr>
</tbody>
</table>

1) 64-bit MATLAB versions are not supported.
2) AutomationDesk’s MATLAB Access library requires MATLAB.
3) Not all MATLAB-related RCP and HIL software products on dSPACE Release 2013-B support 64-bit MATLAB versions. For details, refer to Limitations for Products on 64-bit dSPACE DVD on page 254.
4) Not supported by the RTI FPGA Programming Blockset-FPGA Interface.
5) R2012b, R2013a: Due to performance problems with dSPACE Automotive Simulation Models (ASM), it is recommended to install the following bug fix before you use ASMs with R2012b or R2013a: http://www.mathworks.com/support/bugreports/916069.
6) R2012a: Due to performance problems with dSPACE Automotive Simulation Models (ASM): Opening Bus Selector blocks might not work. Due to performance problems, it is recommended to install the following bug fix before you use ASM with MATLAB R2012a: http://www.mathworks.com/support/bugreports/827771.
7) R2012a: Due to problems with spaces in application path names, it is recommended to install the following bug fix before you use ConfigurationDesk with MATLAB R2012a: http://www.mathworks.com/support/bugreports/814551.

For up-to-date information on additional MATLAB releases which can be used in combination with dSPACE software, refer to http://www.dspace.com/go/sw3rdparty.
Operating System

The following operating systems are supported by the dSPACE products on Release 2013-B:

### 32-bit dSPACE Software
- Windows XP Professional with Service Pack 3 (32-bit version)
- Windows 7 Professional, Ultimate, and Enterprise with Service Pack 1 (32-bit or 64-bit version)
  
  Only the listed editions are supported. The Windows 7 Home and Starter editions are not supported.

### 64-bit dSPACE Software
- Windows 7 Professional, Ultimate, and Enterprise with Service Pack 1 (64-bit version)
  
  Only the listed editions are supported. The Windows 7 Home and Starter editions are not supported.

### Notes and Limitations
- Limitations apply when you use Windows 7. Refer to Limitations for Windows 7 on page 257.
- Support of 64-bit operating systems: 32-bit dSPACE software supports only the 64-bit version of Windows 7. Other 64-bit operating systems (Windows XP and Windows Vista) are not supported.
- 32-bit dSPACE software runs under 64-bit Windows operating systems in a WoW64 (Windows-on-Windows 64-bit) subsystem.
- Limitations apply when you use a 64-bit Windows operating system with 32-bit dSPACE software. Refer to Limitations for 64-Bit Windows Operating Systems in Combination with 32-Bit dSPACE Software on page 254.
- ControlDesk Next Generation can also be installed on the MicroAutoBox Embedded PC, running under Microsoft Windows 7 Ultimate (32-bit version).
- Limitations apply when you use Windows 7. Refer to Limitations for Windows 7 on page 257.
- Only TargetLink, Model Compare, VEOS and various products of the RCP and HIL software package are available as a 64-bit version.
- Limitations apply when you use these 64-bit versions. For details, refer to Limitations for Products on 64-bit dSPACE DVD on page 254.

32-bit dSPACE software running on 64-bit Windows operating systems
dSPACE software runs as a 32-bit application under 64-bit Windows operating systems in a WoW64 (Windows-on-Windows 64-bit) subsystem. WoW64 is the x86 emulator of Windows that allows 32-bit Windows-based applications to run seamlessly on 64-bit versions of Windows. This allows you to use up to 4 GB virtual memory for each 32-bit process. In comparison 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:

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

If you are running third-party firewall software on your host PC, ensure that the TCP/IP communication of dSPACE software is not blocked.

Operating system on dSPACE License Server

If you purchased floating network licenses, you have to install and configure one of the networked PCs as the dSPACE License Server.

The operating system of the dSPACE License Server must be one of the following:

- Windows XP Professional (32-bit version) with Service Pack 3
- Windows Vista Business, Ultimate, or Enterprise (32-bit or 64-bit version) with the latest Service Pack
- Windows 7 Professional, Ultimate, or Enterprise (32-bit or 64-bit version) with the latest Service Pack
- Windows Server 2003 (32-bit or 64-bit version)
- Windows Server 2008 R2 (only available as 64-bit version)
- Windows Server 2012 (only available as 64-bit version)

The dSPACE License Server does not support non-Windows operating systems.
## Run-Time Compatibility of dSPACE Software

### Definition

Run-time compatibility means that:
- dSPACE products can be used in parallel after software installation, even if they are installed in different folders.
- dSPACE products without interaction can run independently of each other.

### Compatibility of products in dSPACE Release 2013-B

dSPACE recommends to use only software products from the same dSPACE Release to provide maximum run-time compatibility.

Note that:
- Limitations regarding run-time compatibility in the dSPACE tool chain might occur when products from different dSPACE Releases are mixed.
  - If dSPACE products interact directly (for example, through automation interfaces) or indirectly (for example, through common file types like A2L) limitations may apply. For major limitations, see below. For minor limitations, refer to the relevant 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 on the patch itself, refer to [http://www.dspace.com/go/CompPatch](http://www.dspace.com/go/CompPatch).
- RCP and HIL software products (on Release 2013-B) cannot be used in combination with RCP and HIL software products from earlier dSPACE releases.

**Major limitations for TargetLink and Model Compare**

The 64-bit version of TargetLink cannot be used in combination with the 32-bit version of Model Compare and vice versa because you can work only with a bit-compatible MATLAB version (32-bit or 64-bit).

**Major limitation for working with a SCALEXIO system**

The products for working with a SCALEXIO system must be compatible. This is only guaranteed for products delivered with the same dSPACE Release. However, it is possible to work with a real-time application created with products from earlier dSPACE releases.

### Combining dSPACE products from earlier releases

For detailed information and notes on the combined use of different products from and with earlier releases, refer to [http://www.dspace.com/go/ds_sw_combi](http://www.dspace.com/go/ds_sw_combi).
### Limitations for 64-Bit Windows Operating Systems in Combination with 32-Bit dSPACE Software

**Objective**

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

**TargetLink: 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. To install the 32-bit version of MATLAB click OK to continue.

### Limitations for Products on 64-bit dSPACE DVD

**Objective**

In general, the 64-bit dSPACE DVD contains the same products as the 32-bit dSPACE DVD. However, the 64-bit DVD contains:

- All MATLAB-related dSPACE products which have been ported to support 64-bit MATLAB versions.
- 32-bit versions of MATLAB-related dSPACE products which were not yet ported to support 64-bit MATLAB versions.

Exception for MATLAB-related products from the RCP and HIL software package: The 64-bit DVD does not contain 32-bit versions of unported MATLAB-related products because these products are not functional in combination with 64-bit MATLAB versions. If you want to use such a product from the RCP and HIL software package, you have to install and use the RCP and HIL software package from the 32-bit DVD.

- All 32-bit dSPACE products that do not relate to MATLAB (e.g., ControlDesk Next Generation).

When using the 64-bit DVD, you should also note the limitations described below.
dSPACE software products ported to 64-bit

The following table lists all dSPACE products which support MATLAB and their availability as 64-bit version on the 64-bit dSPACE DVD.

Keep in mind that dSPACE products without any MATLAB support, e.g., ControlDesk Next Generation, are contained only as 32-bit version.

<table>
<thead>
<tr>
<th>dSPACE Products with MATLAB Support</th>
<th>Contained on 64-bit DVD as ...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64-bit Version</td>
</tr>
<tr>
<td>AutomationDesk</td>
<td>✓</td>
</tr>
<tr>
<td>TargetLink</td>
<td>✓</td>
</tr>
<tr>
<td>Model Compare</td>
<td>✓</td>
</tr>
<tr>
<td>VEOS</td>
<td>✓</td>
</tr>
</tbody>
</table>
### dSPACE Products with MATLAB Support

<table>
<thead>
<tr>
<th>dSPACE Products with MATLAB Support</th>
<th>Contained on 64-bit DVD as ...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64-bit Version</td>
</tr>
<tr>
<td>RCP and HIL software package</td>
<td>✓</td>
</tr>
<tr>
<td>RTi and RTi-MP</td>
<td>✓</td>
</tr>
<tr>
<td>RTi Gigalink Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi CAN Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi CAN MultiMessage Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi LIN MultiMessage Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi FlexRay Configuration Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi FPGA Programming Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi Ethernet UDP Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi XCP on Ethernet Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi Watchdog Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi RapidPro Control Unit Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>RTi Bypass Blockset</td>
<td>– 2)</td>
</tr>
<tr>
<td>RTi AUTOSAR Package</td>
<td>– 2)</td>
</tr>
<tr>
<td>MLIB/MTRACE</td>
<td>– 3)</td>
</tr>
<tr>
<td>ConfigurationDesk</td>
<td>–</td>
</tr>
<tr>
<td>FlexRay Configuration Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>FlexRay Configuration Tool</td>
<td>✓</td>
</tr>
<tr>
<td>ModelDesk</td>
<td>–</td>
</tr>
<tr>
<td>Automotive Simulation Models</td>
<td>✓</td>
</tr>
<tr>
<td>MotionDesk</td>
<td>–</td>
</tr>
<tr>
<td>MotionDesk Blockset</td>
<td>✓</td>
</tr>
<tr>
<td>Flight Rec Data Merger</td>
<td>–</td>
</tr>
<tr>
<td>Further products of RCP and HIL</td>
<td>–</td>
</tr>
</tbody>
</table>

1) The 64-bit version is planned for future releases.
2) Support of 64-bit MATLAB versions is planned for future releases. Use the 32-bit DVD (with 32-bit MATLAB version support) to install this product.
3) Support of 64-bit MATLAB versions is not planned for future releases. Use the 32-bit DVD (with 32-bit MATLAB version support) to install this product.
4) Supports 64-bit MATLAB versions.
Further product-specific limitations

**MAT file support** The following products only support reading and writing MAT files of file format version 5.0. MAT files of this version can be created in MATLAB by using the `save` command with option `-v6`:
- ModelDesk (Maneuver Editor, Road Generator)
- ControlDesk Next Generation (ControlDesk 5.1)
- AutomationDesk 4.0 (MATLAB Access Library)
- The matlablib2 Python modules

**Automation of 64-bit MATLAB versions** AutomationDesk 4.0 (MATLAB Access Library), the matlablib2 Python modules do not support the automation of the 64-bit MATLAB versions.

**RTI-MP** The command `rtimpdiag` is not functional. This command is based on MLIB, which will not be provided for MATLAB (64-bit).

**Parallel installation of a 32-bit and a 64-bit version of MATLAB** On a 64-bit operating system, you can install both a 32-bit and a 64-bit version of a particular MATLAB release, e.g., R2013b. However, MATLAB release versions share the same preference settings folder, which means that you have to set all architecture-dependent settings such as the MEX compiler settings each time you switch between different versions of one MATLAB release.

Limitations for the 64-bit version of TargetLink

**Importing an A2L file** It is not possible to import an A2L file into a 64-bit version of TargetLink. However, you can use a workaround described in Basics of Importing ASAM MCD-2 MC Files ([TargetLink Data Dictionary A2L Import and Export](#)).

Limitations for Windows 7

**Objective** Some limitations apply when you use 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.
## Compatibility Information

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Closing dSPACE software before PC shutdown</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>User Account Control</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>USB devices</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Using boards with ISA interface installed in the host PC</strong></td>
<td>When installed directly in the host PC, ISA boards such as the DS813 cannot be used with Windows 7 with the standard installation routines. If necessary, contact dSPACE Support.</td>
</tr>
</tbody>
</table>
| **Limitation for VPUs built with VEOS 3.1 using the GCC compiler** | You cannot download VPUs built with dSPACE Target for Offline Simulation in VEOS 3.1 and the GCC compiler with the following operating systems:  
- Windows 7 SP1 (32-bit version)  
- Windows 7 SP1 (64-bit version)  

**Workarounds:**  
- Install the KB2872339 security update for Windows 7. This update is helpful only for Windows 7 SP1 (64-bit version), however.  
- Use the MSVC compiler (instead of the GCC compiler) to build VPUs with dSPACE Target for Offline Simulation in VEOS 3.1.
Numerics

<table>
<thead>
<tr>
<th>Numerics</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-bit dSPACE DVD</td>
</tr>
<tr>
<td>limitations 254</td>
</tr>
<tr>
<td><strong>A</strong></td>
</tr>
<tr>
<td>ASM Base InCylinder Blockset</td>
</tr>
<tr>
<td>migrating 63</td>
</tr>
<tr>
<td>new features 63</td>
</tr>
<tr>
<td>ASM blocksets</td>
</tr>
<tr>
<td>migrating 62</td>
</tr>
<tr>
<td>ASM Diesel Engine Blockset</td>
</tr>
<tr>
<td>migrating 66</td>
</tr>
<tr>
<td>new features 65</td>
</tr>
<tr>
<td>ASM Diesel Exhaust Blockset</td>
</tr>
<tr>
<td>migrating 69</td>
</tr>
<tr>
<td>new features 67</td>
</tr>
<tr>
<td>ASM Diesel InCylinder Blockset</td>
</tr>
<tr>
<td>migrating 70</td>
</tr>
<tr>
<td>ASM Drivetrain Basic Blockset</td>
</tr>
<tr>
<td>migrating 71</td>
</tr>
<tr>
<td>ASM Electric Components Blockset</td>
</tr>
<tr>
<td>migrating 72</td>
</tr>
<tr>
<td>ASM Engine Gasoline Blockset</td>
</tr>
<tr>
<td>new features 75</td>
</tr>
<tr>
<td>ASM Environment Blockset</td>
</tr>
<tr>
<td>migrating 74</td>
</tr>
<tr>
<td>new features 74</td>
</tr>
<tr>
<td>ASM Gasoline InCylinder Blockset</td>
</tr>
<tr>
<td>migrating 76</td>
</tr>
<tr>
<td>new features 76</td>
</tr>
<tr>
<td>ASM InCylinder Blockset</td>
</tr>
<tr>
<td>migrating 79</td>
</tr>
<tr>
<td>new features 79</td>
</tr>
<tr>
<td>ASM Truck Blockset</td>
</tr>
<tr>
<td>migrating 81</td>
</tr>
<tr>
<td>new features 80</td>
</tr>
<tr>
<td>ASM Turbocharger Blockset</td>
</tr>
<tr>
<td>migrating 82</td>
</tr>
<tr>
<td>new features 82</td>
</tr>
<tr>
<td>ASM Vehicle Dynamics Blockset</td>
</tr>
<tr>
<td>migrating 85</td>
</tr>
<tr>
<td>new features 84</td>
</tr>
<tr>
<td>AutomationDesk</td>
</tr>
<tr>
<td>migration 55</td>
</tr>
<tr>
<td>new features 49</td>
</tr>
<tr>
<td>AUTOSAR</td>
</tr>
<tr>
<td>TargetLink related migration</td>
</tr>
<tr>
<td>236</td>
</tr>
<tr>
<td><strong>C</strong></td>
</tr>
<tr>
<td>Common Program Data folder</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>CommonProgramDataFolder</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>ControlDesk Next Generation</td>
</tr>
<tr>
<td>migration 113</td>
</tr>
<tr>
<td>new features 98</td>
</tr>
<tr>
<td><strong>D</strong></td>
</tr>
<tr>
<td>DCI Configuration Tool</td>
</tr>
<tr>
<td>new features 123</td>
</tr>
<tr>
<td>Documents folder 13</td>
</tr>
<tr>
<td>DocumentsFolder 13</td>
</tr>
<tr>
<td>dSPACE ECU Flash Programming Tool migration 125</td>
</tr>
<tr>
<td>dSPACE FlexRay Configuration Package migration 131</td>
</tr>
<tr>
<td>new features 129</td>
</tr>
<tr>
<td>dSPACE HIL API .NET migration 127</td>
</tr>
<tr>
<td>dSPACE Python Extensions migration 134</td>
</tr>
<tr>
<td>new features 133</td>
</tr>
<tr>
<td><strong>E</strong></td>
</tr>
<tr>
<td>ECU Interface Manager</td>
</tr>
<tr>
<td>migration 140</td>
</tr>
<tr>
<td>new features 137</td>
</tr>
<tr>
<td><strong>F</strong></td>
</tr>
<tr>
<td>Firmware Manager</td>
</tr>
<tr>
<td>new features 141</td>
</tr>
<tr>
<td><strong>G</strong></td>
</tr>
<tr>
<td>general enhancements and changes</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td><strong>H</strong></td>
</tr>
<tr>
<td>host PC software</td>
</tr>
<tr>
<td>operating system 251</td>
</tr>
<tr>
<td><strong>K</strong></td>
</tr>
<tr>
<td>key features 26</td>
</tr>
<tr>
<td><strong>L</strong></td>
</tr>
<tr>
<td>Limitations</td>
</tr>
<tr>
<td>TargetLink obsolete limitations 242</td>
</tr>
<tr>
<td>limitations for products on 64-bit dSPACE DVD 254</td>
</tr>
<tr>
<td>limitations for Windows 64-Bit and dSPACE 64-Bit software 254</td>
</tr>
<tr>
<td>limitations for Windows 7 257</td>
</tr>
<tr>
<td>Local Program Data folder 13</td>
</tr>
<tr>
<td>LocalProgramDataFolder 13</td>
</tr>
<tr>
<td><strong>M</strong></td>
</tr>
<tr>
<td>MATLAB</td>
</tr>
<tr>
<td>supported releases 250</td>
</tr>
<tr>
<td><strong>MicroAutoBox</strong></td>
</tr>
<tr>
<td>new features 155</td>
</tr>
<tr>
<td>migrating</td>
</tr>
<tr>
<td>ASM Base InCylinder Blockset</td>
</tr>
<tr>
<td>63</td>
</tr>
<tr>
<td>ASM blocksets</td>
</tr>
<tr>
<td>62</td>
</tr>
<tr>
<td>ASM Diesel Engine Blockset</td>
</tr>
<tr>
<td>66</td>
</tr>
<tr>
<td>ASM Diesel Exhaust Blockset</td>
</tr>
<tr>
<td>69</td>
</tr>
<tr>
<td>ASM Diesel InCylinder Blockset</td>
</tr>
<tr>
<td>70</td>
</tr>
<tr>
<td>ASM Drivetrain Basic Blockset</td>
</tr>
<tr>
<td>71</td>
</tr>
<tr>
<td>ASM Environment Blockset</td>
</tr>
<tr>
<td>74</td>
</tr>
<tr>
<td>ASM Gasoline InCylinder Blockset</td>
</tr>
<tr>
<td>76</td>
</tr>
<tr>
<td>ASM Trailer Blockset</td>
</tr>
<tr>
<td>79</td>
</tr>
<tr>
<td>ASM Truck Blockset</td>
</tr>
<tr>
<td>81</td>
</tr>
<tr>
<td>ASM Turbocharger Blockset</td>
</tr>
<tr>
<td>82</td>
</tr>
<tr>
<td>ASM Vehicle Dynamics Blockset</td>
</tr>
<tr>
<td>85</td>
</tr>
<tr>
<td>migration</td>
</tr>
<tr>
<td>AutomationDesk</td>
</tr>
<tr>
<td>55</td>
</tr>
<tr>
<td>new features 49</td>
</tr>
<tr>
<td><strong>MotionDesk</strong></td>
</tr>
<tr>
<td>migration 147</td>
</tr>
<tr>
<td>new features 145</td>
</tr>
<tr>
<td><strong>MicroAutoBox</strong></td>
</tr>
<tr>
<td>migration 150</td>
</tr>
<tr>
<td>new features 149</td>
</tr>
</tbody>
</table>
new features

ASM Gasoline InCylinder Blockset 76
ASM Trailer Blockset 78
ASM Truck Blockset 80
ASM Turbocharger Blockset 82
ASM Vehicle Dynamics Blockset 84
AutomationDesk 49
ControlDesk Next Generation 98
DCI Configuration Tool 123
dSPACE FlexRay Configuration Package 129
dSPACE Python Extensions 133
ECU Interface Manager 137
Firmware Manager 141
MicroAutoBox 155
Model Compare 143
ModelDesk 145
MotionDesk 149
RTI Bypass Blockset 163
RTI CAN MultiMessage Blockset 167
RTI FPGA Programming Blockset 173
RTI LIN MultiMessage Blockset 171
RTI Watchdog Blockset 179
RTI/RTI-MP 153
RTLib 153
SystemDesk 182
VEOS 245

not supported MATLAB features 153

P
product overview 22

R
Real-Time Testing
migration 151
requirements
host PC software
operating system 251
RTI Bypass Blockset
migration 166
new features 163
RTI CAN MultiMessage Blockset
migration 169
new features 167
RTI FPGA Programming Blockset
migration 176
new features 173
RTI LIN MultiMessage Blockset
migration 172
new features 171
RTI Watchdog Blockset
migration 180
new features 179
RTI/RTI-MP
new features 153

RTLib
new features 153

S
supported MATLAB releases 250
system requirements
operating system 251
SystemDesk
new features 182

T
TargetLink
API commands
changes 240
AUTOSAR features, new
component-based
development 205
hook functions, new 221
multiple instantiation of SWCs 203
supported releases 202
code changes
migration 232
code efficiency, improvement of
196
documentation changes
migration 238
documentation improvements 216
hook functions
AUTOSAR 221
new 221
migrating to
new version 222
migration
AUTOSAR related 236
code changes 232
documentation changes 238
obsolete limitations 242
various aspects 239
new API commands
V-ECU generation 220
virtual ECU generation 220
new code generator options 211
new features 192
enumeration support 193
general changes 219
general enhancements 219
improved Stateflow support 195
Improved System Preparation 194
new version
migrating to 222
newly supported Simulink blocks 192
target support
discontinued compiler versions 210
discontinued evaluation boards 210
new compiler versions 210
new evaluation boards 210
supported targets 210

TargetLink Data Dictionary
API commands
changes 240
new commands 201
discontinued features 241
migrating to
new version 222
migration 225
discontinued documentation 225
upgrading existing data
dictionaries 228, 229
upgrading from version 2.x 231
new features 192
new version
migrating to 222

V
VEOS
new features 245
version history 22

W
Windows 64-Bit
limitations 254
Windows 7
limitations 257