New Features and Migration

TargetLink 2.2 – December 2006
How to Contact dSPACE

Mail: dSPACE GmbH
    Technologiepark 25
    33100 Paderborn
    Germany
Tel.: +49 52 51 16 38-0
Fax: +49 52 51 6 65 29
E-mail: info@dspace.de
Web: http://www.dspace.com
General Technical Support: support@dspace.de
    +49 52 51 16 38-941
    http://www.dspace.com/goto?support
TargetLink Support: support.tl@dspace.de
    +49 52 51 16 38-700

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/goto?support for software updates and patches.

Important Notice

This document contains proprietary information that is protected by copyright. All rights are reserved. Neither the documentation nor software may 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 2006 by:
dSPACE GmbH
Technologiepark 25
33100 Paderborn
Germany

This publication and the contents hereof are subject to change without notice.
ConfigurationDesk, ControlDesk, TargetLink and CalDesk 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** 5
- Accessing Online Help and PDF Files ........................................... 6
- Document Symbols and Conventions ........................................... 7
- Related Documents ................................................................. 8

**New Features of TargetLink 2.2** 11
- General New Features .............................................................. 12
- Supported MATLAB Platforms ................................................ 12
- Combination of dSPACE Products ............................................. 14
- TargetLink 2.2 Compatibility with MATLAB, and RCP and HIL Software .............................................................. 15
- New Key Features ................................................................. 16
- New Installation Features ....................................................... 16
- New Production Code Generation Features .............................. 18
- AUTOSAR Software Component Support ................................. 18
- Pointer To Structures at Function Interfaces .............................. 20
- Target Support ................................................................. 22
- Model-Linked Code View ....................................................... 23
- Graphical User Interface for Code Generator Options ................. 24
- Inheritance and Propagation of Signal Properties ...................... 26
- Changes on Autoscaling ....................................................... 27
- TargetLink Blocks Supporting Bus Signals .............................. 28
- Common TargetLink Context Menu Options ............................. 28
- Enhancements Regarding Stateflow ......................................... 29
- Miscellaneous Features .......................................................... 29

**New Features of dSPACE Data Dictionary 1.4** 33
- New Key Features ................................................................. 33
- General Changes in the dSPACE Data Dictionary ...................... 35
- Changes in the Data Dictionary Manager .................................... 36
- New and Modified DD API Commands ...................................... 39
- Changes in the ASAM-MCD 2MC Import and Export Module ........ 41

**Migrating to TargetLink 2.2** 43
- Default Behavior of the Code Generator ................................. 43
- Simulation Behavior ............................................................. 47
Contents

Build Process ........................................................................... 47

Migrating to dSPACE Data Dictionary 1.4 49
Upgrading Data Dictionary Project Files ................................... 50
How to Upgrade DD Project Files via the Data Dictionary Manager ................................................................. 52
Changed or Obsolete DD API Commands ................................. 53
Adapting the Target_Info file for A2L File Export .................. 54

Index 55
About This Document

Objective
This document provides you with a brief overview of the major new features of TargetLink 2.2 since the previous release.
dSPACE Release 5.2 contains all the software from dSPACE on one DVD:
- RCP and HIL software
- CalDesk 1.4
- TargetLink 2.2
This document does not describe the new features of RCP and HIL software and CalDesk 1.4.

Where to go from here
Information in this section

| Accessing Online Help and PDF Files | 6 |
| Document Symbols and Conventions    | 7 |
| Related Documents                   | 8 |
Accessing Online Help and PDF Files

Objective

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

Online help

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

- **Windows Start menu**
  - Click Start – Programs – dSPACE Tools – dSPACE HelpDesk.
- **Context-sensitive**
  - Press the F1 key or click the Help button in the dSPACE software.
- **Local installation on your host PC**
  - Double-click the dSPACEHelpDesk.chm file in %DSPACE_ROOT%\Doc\Online.

PDF Files

You can access the PDF files as follows:

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

    ![PDF Link Example]

    - **Local installation on your host PC**
      - Double-click the PDF file in %DSPACE_ROOT%\Doc\Print.
## Document Symbols and Conventions

### Symbols

The following symbols are used in this document:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Indicates a general hazard that may cause personal injury of any kind if you do not avoid it by following the instructions given.</td>
</tr>
<tr>
<td>⚡</td>
<td>Indicates the danger of electric shock which may cause death or serious injury if you do not avoid it by following the instructions given.</td>
</tr>
<tr>
<td>⚠</td>
<td>Indicates a hazard that may cause material damage if you do not avoid it by following the instructions given.</td>
</tr>
<tr>
<td>🗣️</td>
<td>Indicates important information that you must note to avoid malfunctions.</td>
</tr>
<tr>
<td>💡</td>
<td>Indicates tips containing useful information to make your work easier.</td>
</tr>
</tbody>
</table>

### Naming conventions

The following abbreviations and formats are used in this document:

- **%name%** Names enclosed in percent signs refer to environment variables for file and path names, for example, `%DSPACE_ROOT%` specifies the location of your dSPACE installation in the file system.

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

- ** collegiate** 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.
# Related Documents

**Recommended reading**

You are recommended to read the following documents when working with TargetLink:

## Information in other documents

<table>
<thead>
<tr>
<th>TargetLink guides</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Software Installation and Management Guide</strong></td>
<td>Provides detailed instructions on installing and handling the dSPACE software. It also shows you how to manage dSPACE licenses.</td>
</tr>
<tr>
<td><strong>TargetLink Production Code Generation Guide</strong></td>
<td>Outlines the typical code generation workflow and introduces you to the basic features provided by TargetLink. It also describes how DD objects (dSPACE Data Dictionary objects) are used at TargetLink blocks.</td>
</tr>
<tr>
<td><strong>TargetLink Advanced Practices Guide</strong></td>
<td>Describes advanced features and blocks you can use to optimize the generated code, embed user-defined code and adapt company-specific coding styles.</td>
</tr>
<tr>
<td><strong>TargetLink Blockset Guide</strong></td>
<td>Explains how to use the TargetLink Blockset in stand-alone mode.</td>
</tr>
<tr>
<td><strong>TargetLink Multirate Modeling Guide</strong></td>
<td>Provides information on the concepts and techniques that you can apply to generate code for multirate systems and explains how to develop OSEK-compliant multitasking applications.</td>
</tr>
<tr>
<td><strong>TargetLink AUTOSAR Modeling Guide</strong></td>
<td>Explains how to model and generate code for AUTOSAR-compliant software components.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TargetLink references</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TargetLink Block and Object Reference</strong></td>
<td>Describes TargetLink blocks, supported Simulink blocks and Stateflow objects.</td>
</tr>
<tr>
<td><strong>TargetLink Tool and Utility Reference</strong></td>
<td>Provides information on the TargetLink tools and utilities.</td>
</tr>
<tr>
<td><strong>TargetLink API Reference</strong></td>
<td>Describes the available API commands and provides information on custom look-up functions.</td>
</tr>
</tbody>
</table>
TargetLink Target Reference
Provides concise information on the target boards and compilers supported by TargetLink.

TargetLink File and Message Reference
Lists all the files used in a production code target application and error messages according to category.

dSPACE Data Dictionary

dSPACE Data Dictionary Basic Concepts Guide
Introduces you to the features provided by the dSPACE Data Dictionary.

dSPACE Data Dictionary Manager Reference
Provides detailed information on the menus, context menus, and dialogs contained in the Data Dictionary Manager.

dSPACE Data Dictionary XML Import and Export
Introduces you to the XML import and export module of the dSPACE Data Dictionary.

dSPACE Data Dictionary ASAM-MCD 2MC Import and Export
Introduces you to the ASAM-MCD 2MC import and export module of the dSPACE Data Dictionary.

dSPACE Data Dictionary OIL Import and Export
Introduces you to the OIL import and export module of the dSPACE Data Dictionary.

dSPACE Data Dictionary MATLAB API Reference
Provides detailed information on the features of the Data Dictionary MATLAB API.

dSPACE Data Dictionary Data Model Reference
Provides detailed information on the data model of the dSPACE Data Dictionary.
# New Features of TargetLink 2.2

**Objective**

TargetLink 2.2 has the following new features, enhancements and changes.

**Where to go from here**

Information in this section

<table>
<thead>
<tr>
<th>General New Features</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Production Code Generation Features</td>
<td>18</td>
</tr>
</tbody>
</table>
General New Features

Information on general new features of TargetLink 2.2 is provided below.

Where to go from here

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported MATLAB Platforms</td>
<td>12</td>
</tr>
<tr>
<td>Combination of dSPACE Products</td>
<td>14</td>
</tr>
<tr>
<td>TargetLink 2.2 Compatibility with MATLAB, and RCP and HIL Software</td>
<td>15</td>
</tr>
<tr>
<td>New Key Features</td>
<td>16</td>
</tr>
<tr>
<td>New Installation Features</td>
<td>16</td>
</tr>
</tbody>
</table>

Supported MATLAB Platforms

TargetLink 2.2 supports the following MATLAB releases:

<table>
<thead>
<tr>
<th>MATLAB Release...</th>
<th>Is Supported by dSPACE Release 5.2 Software Item...</th>
<th>TargetLink 2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB R2006b1)</td>
<td>RCP and HIL Software</td>
<td>Yes</td>
</tr>
<tr>
<td>MATLAB R2006a+1)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MATLAB R14SP3</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MATLAB R14SP2+</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MATLAB R14SP2</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MATLAB R13SP2</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MATLAB R13SP1+</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

1) Not supported by RTI FlexRay Blockset.
New Features of TargetLink 2.2

TargetLink compatibility with Java Virtual Machine

TargetLink needs a specific Java Virtual Machine (JVM):

- JVM 1.5.0 originally installed with MATLAB R14SP2 or later causes errors that particularly impair work with TargetLink, although the same errors also occur without TargetLink. Thus, TargetLink 2.2 has been released explicitly for Java Virtual Machine 1.4.2_08.

  It is strongly recommended to use this JVM in conjunction with TargetLink 2.2.

- You can install the JVM 1.4.2_08 from Tools\Java in the root folder of the DVD. To make MATLAB use this JVM, you must set the MATLAB_JAVA environment variable to the path of the JVM 1.4.2_08.

For information on how to install TargetLink and a detailed description of its requirements, refer to Installation Overview.

Related topics

<table>
<thead>
<tr>
<th>Basics</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB Compatibility Information</td>
</tr>
</tbody>
</table>

TargetLink New Features and Migration  December 2006  13
## Combination of dSPACE Products

To install a dSPACE Release together with CalDesk or TargetLink in the **same folder**, you have to install the products in a certain order. Combinations that do not comply with these installation orders will not work properly.

<table>
<thead>
<tr>
<th>Combination Is Possible</th>
<th>After dSPACE Software ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>When Installing dSPACE Software ...</td>
<td>dSPACE Release</td>
</tr>
<tr>
<td></td>
<td>5.0</td>
</tr>
<tr>
<td>dSPACE Release 5.0</td>
<td>-</td>
</tr>
<tr>
<td>dSPACE Release 5.1</td>
<td>-</td>
</tr>
<tr>
<td>dSPACE Release 5.2¹</td>
<td>-</td>
</tr>
<tr>
<td>TargetLink 2.0.6/2.0.7</td>
<td>No</td>
</tr>
<tr>
<td>TargetLink 2.1.6</td>
<td>Yes</td>
</tr>
<tr>
<td>TargetLink 2.2</td>
<td>No</td>
</tr>
<tr>
<td>CalDesk 1.3</td>
<td>Yes</td>
</tr>
<tr>
<td>CalDesk 1.4</td>
<td>Yes</td>
</tr>
</tbody>
</table>

¹ Here, “dSPACE Release 5.2” stands for the RCP and HIL software of the release, not including TargetLink 2.2 and CalDesk 1.4.


The installations are listed in the dSPACE Installation Manager and the **Add/Remove Programs** list of the Windows Control Panel. Since installations are listed alphabetically and not chronologically, do not forget to make a note of the order in which you combined the dSPACE software in. Otherwise, it will be difficult to reconstruct the installation order later on.
TargetLink 2.2 Compatibility with MATLAB, and RCP and HIL Software

<table>
<thead>
<tr>
<th>Objective</th>
<th>TargetLink has special requirements regarding compatibility with MATLAB, and RCP and HIL software. CalDesk is not considered here as it does not require MATLAB.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCP and HIL software and MATLAB dependencies</td>
<td>The table below shows you which MATLAB releases, and RCP and HIL software, you can install TargetLink 2.2 with. For combinations that are not mentioned explicitly in the table below, the installation of TargetLink 2.2 is not possible.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Installing TargetLink 2.2 Is Possible with MATLAB Release...</th>
<th>In Combination with RCP and HIL Software Contained in...</th>
<th>dSPACE Release 5.1</th>
<th>dSPACE Release 5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2006b</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>R2006a+</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>R14SP3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>R14SP2+</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>R14SP2</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>R13SP2</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>R13SP1+</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

The same applies to the TargetLink Blockset 2.2 (stand-alone).

dSPACE Data Dictionary Manager | You can install TargetLink 2.2 with dSPACE Data Dictionary Manager version 1.4.
New Features of TargetLink 2.2

New Key Features

<table>
<thead>
<tr>
<th>Objective</th>
<th>Information on the new key features of TargetLink 2.2 is provided below.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetLink AUTOSAR Blockset</td>
<td>TargetLink 2.2 includes blocks for implementing software components (SWCs) compliant with the AUTOSAR standard. Refer to AUTOSAR Software Component Support on page 18.</td>
</tr>
<tr>
<td>Function interfaces with pointers to structures</td>
<td>TargetLink 2.2 now supports pointers to structures in the function signature, which is particularly efficient if there are a large number of function parameters, for example. Refer to Pointer To Structures at Function Interfaces on page 20.</td>
</tr>
<tr>
<td>(New) microcontrollers, boards, and compilers</td>
<td>TargetLink 2.2 supports new microcontrollers, boards, and compiler versions. Refer to Target Support on page 22.</td>
</tr>
</tbody>
</table>

New Installation Features

<table>
<thead>
<tr>
<th>New Installation Features</th>
<th>All the dSPACE software products are shipped on one DVD. The DVD for dSPACE Release 5.2 contains the following dSPACE items:</th>
</tr>
</thead>
</table>
|                         | ■ RCP and HIL software  
|                         | ■ TargetLink 2.2  
|                         | ■ CalDesk 1.4 |

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

The DVD contains four setups for the dSPACE software. You can handle various installation scenarios with the installation programs on the DVD. The installation programs will guide you through the installation of the software. Installing the software is convenient and quick, especially for users who want to install a combination of dSPACE software items, for example, RCP/HIL software and CalDesk 1.4, because they do not need to install using several CDs or DVDs.
New documentation for software installation

As of dSPACE Release 5.2, dSPACE provides information about software installation in the following documents:

- The *Quick Software Installation Guide* describes the software installation from scratch. It also guides you to further basic information and advanced practices in other documents.

  The guide is available in printed form and as a printable PDF file (SWQuickInst.pdf) in the root folder of the DVD.

- Unlike the *Quick Software Installation Guide*, the *Software Installation and Management Guide* contains all the information you need for installing and managing dSPACE software. In addition to basic practices, it gives detailed information on advanced practices, for example, managing dSPACE software and managing dSPACE licenses.

  The guide is available as a printable PDF file (SWInst.pdf) in the root folder of the DVD.
New Production Code Generation Features

Where to go from here

<table>
<thead>
<tr>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR Software Component Support</td>
</tr>
<tr>
<td>Pointer To Structures at Function Interfaces</td>
</tr>
<tr>
<td>Target Support</td>
</tr>
<tr>
<td>Model-Linked Code View</td>
</tr>
<tr>
<td>Graphical User Interface for Code Generator Options</td>
</tr>
<tr>
<td>Inheritance and Propagation of Signal Properties</td>
</tr>
<tr>
<td>Changes on Autoscaling</td>
</tr>
<tr>
<td>TargetLink Blocks Supporting Bus Signals</td>
</tr>
<tr>
<td>Common TargetLink Context Menu Options</td>
</tr>
<tr>
<td>Enhancements Regarding Stateflow</td>
</tr>
<tr>
<td>Miscellaneous Features</td>
</tr>
</tbody>
</table>

AUTOSAR Software Component Support

TargetLink AUTOSAR blockset

TargetLink 2.2 includes blocks for implementing software components (SWCs) compliant with the AUTOSAR standard (see www.autosar.org). These components contain the actual function code for an electronic control unit (ECU). The specifications required for the SWCs are made in the familiar TargetLink/Simulink® environment, which makes the modeling of AUTOSAR SWCs particularly efficient.

TargetLink 2.2 not only generates code for AUTOSAR SWCs, but also gives extensive support for modeling and simulating them. Support for AUTOSAR SWCs includes:

- Modeling and simulating AUTOSAR software components.
- Code generation for AUTOSAR software components.
- AUTOSAR SWC description import and export.
Modeling and simulating AUTOSAR software components

Using the TargetLink AUTOSAR blocks, you can model AUTOSAR-compliant SWCs, with runnables and ports.

An SWC has runnables and ports. The ports of an SWC are used for communication, and runnables implement the internal behavior that represents the functionality of the SWC. TargetLink supports the two AUTOSAR communication concepts, sender-receiver communication and client-server communication.

Simulation of SWCs is supported for MIL, SIL, and PIL simulation mode, so you can analyze the SWC’s behavior on your host PC and the target hardware.
New Features of TargetLink 2.2

For further details, refer to the TargetLink AUTOSAR Modeling Guide.

**Code generation for AUTOSAR software components**

After you have finished the specification work, you can generate AUTOSAR-compliant code. The runnables are generated as C functions containing run-time environment (RTE) macros that are needed for data exchange.

**AUTOSAR SWC description import and export**

In addition to generating the actual C code, TargetLink 2.2 also provides a standardized description of the AUTOSAR software components (SWC description) in XML format. This description is needed for integration of the SWC code in an ECU executable, as it contains information on SWC elements such as runnables and ports. The SWC description is compatible with the AUTOSAR SWC template.

TargetLink lets you import and export SWC descriptions (SWC-Ds) to the dSPACE Data Dictionary. The dSPACE Data Dictionary Manager also has an integrated AUTOSAR SWC-D Merge Explorer, which lets you copy selected elements to the currently open DD project file or generate a difference report.

**Pointer To Structures at Function Interfaces**

**Function interfaces with pointers to structures**

TargetLink 2.2 now supports pointers to structures in the function signature, which is particularly efficient if there are a large number of function parameters, for example.

Components of structures which are passed to a function as pointer arguments can be used for function input, output, and internal block variables, and also for parameter and state variables, and Stateflow variables. The components of a structure can, for example, be passed as shown below:
New Features of TargetLink 2.2

TL_SubsystemFunction(input_type *input_struct, para_type *param_struct, output_type *output_struct)

A common and thus important use case for passing function input variables via pointers to structures is when Simulink signal buses enter a TargetLink subsystem or a subsystem containing a TargetLink Function block.

Pointers to structures in the function signature are especially useful to:

- Avoid global variables for parameter passing (improved structuring capabilities for the generated production code).
- Create concise function signatures, which otherwise have long argument lists.
- Minimize the stack consumption

**Specification methods**

You can use different methods to let TargetLink create pointers to structures in the function signature during code generation:

- Create a structure using a variable class where `scope` is set to `‘ref_param’`, and then reference the structure components from the respective block dialogs.
- Configure a bus signal using bus port blocks so that during code generation the signal is generated as an implicit structure, and so that the function signature of the subsystem that the bus signal is passed to is given a pointer to the structure. For these purposes, you have to specify the settings shown in the illustration below in the relevant bus port blocks:
For details on how to specify pointers to structures, refer to *Structuring Production Code Using Pointers to Structs* (TargetLink Advanced Practices Guide).

## Target Support

The following table shows the combinations of microcontrollers, boards, and compilers as supported by TargetLink 2.2 (TargetLink abbreviations). New microcontrollers, boards, and compiler versions are underlined. For details, refer to TargetLink Target Reference.

<table>
<thead>
<tr>
<th>Microcontroller Family</th>
<th>Board</th>
<th>Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>C16x</td>
<td>Promo167</td>
<td>Task60, Task75, Task80, Task85, Task86</td>
</tr>
<tr>
<td>ST10F276</td>
<td>START276</td>
<td>Task75, Task80, Task85, Task86</td>
</tr>
<tr>
<td>H8S/26xx</td>
<td>EVB2633F</td>
<td>Hit30, Hit60</td>
</tr>
<tr>
<td>HCS12</td>
<td>HCS12EVB</td>
<td>Cosmic44, Cosmic45, Cosmic46, Cosmic47</td>
</tr>
<tr>
<td></td>
<td>HCS12DPS12EVB</td>
<td>Met12, Met20, Met31</td>
</tr>
<tr>
<td>S12X</td>
<td>S12XEVB</td>
<td>Cosmic46, Cosmic47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Met41, Met45</td>
</tr>
<tr>
<td>M32R</td>
<td>MSA2114</td>
<td>Mcc32r20, Mcc32r43, Mcc32r50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gau013, Gau0108</td>
</tr>
<tr>
<td>MPC5xx</td>
<td>CME555</td>
<td>Diab43, Diab54, Diab56, Diab57, Diab58</td>
</tr>
<tr>
<td></td>
<td>CMD565</td>
<td>GHS30, GHS35, GHS36, GHS34, GHS42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Met60, Met81, Met85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPC5xx</td>
<td>MPC5554DEMO</td>
<td>Diab52, Diab53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GHS40, GHS42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Met15, GNU34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OS1603</td>
<td>Diab53, M8152</td>
</tr>
<tr>
<td>NEC V850ES</td>
<td>DL_V850F3239</td>
<td>GHS35, GHS34, GHS32, GHS40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NEC25, NEC27, NEC31</td>
</tr>
<tr>
<td>SH2</td>
<td>SH2eEVB</td>
<td>Hit50, Hit51, Hit60, Hit70, Hit80, Hit90</td>
</tr>
<tr>
<td></td>
<td>EVB7058</td>
<td></td>
</tr>
<tr>
<td>TI TMS440R1x</td>
<td>EVB470R1</td>
<td>Tccs13, Tccs21, Tccs22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriCore1775</td>
<td>TBTC1775</td>
<td>Task11, Task13, Task14, Task15, Task22, Task23</td>
</tr>
<tr>
<td>TriCore1796</td>
<td>TBTC1796</td>
<td>Task22, Task23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GNU33</td>
</tr>
</tbody>
</table>
Discontinued boards and compiler versions

<table>
<thead>
<tr>
<th>Discontinued boards:</th>
<th>Discontinued boards:</th>
</tr>
</thead>
<tbody>
<tr>
<td>HC12EVB (HC12 microcontroller family)</td>
<td>□ HC12EVB (HC12 microcontroller family)</td>
</tr>
<tr>
<td>EVB2655 (H8S/26xx microcontroller family)</td>
<td>□ EVB2655 (H8S/26xx microcontroller family)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discontinued compiler versions:</th>
<th>Discontinued compiler versions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task51 for Promo167 evaluation board (C16x microcontroller family)</td>
<td>□ Task51 for Promo167 evaluation board (C16x microcontroller family)</td>
</tr>
</tbody>
</table>

**Model-Linked Code View**

**Tracing Generated Code Back to the Model**

When performing code reviews, it is helpful and convenient to trace variables, types, and functions, and to quickly navigate from the generated code back to the relevant block or Stateflow element in the model, or to the dSPACE Data Dictionary (DD).

You can let TargetLink generate a model-linked code view file via the **Generate model-linked code view** option on the Advanced page of the TargetLink Main Dialog. This file is an HTML representation of the generated production code, containing hyperlinks that let you navigate to the blocks or subsystems from which the code was generated.
During code generation, the TargetLink Code Generator not only generates header and code files, but also an HTML version for each of the files. After starting the MATLAB Web Browser, you can browse all HTML files that reside in the CodeViewFiles subfolder of the model that is currently open. Clicking a hyperlink lets you select the destination you want to jump to, i.e., the source that code generation was based on. The following code fragment illustrates a hyperlink and its possible destinations:

```plaintext
 During code generation, the TargetLink Code Generator not only generates header and code files, but also an HTML version for each of the files. After starting the MATLAB Web Browser, you can browse all HTML files that reside in the CodeViewFiles subfolder of the model that is currently open. Clicking a hyperlink lets you select the destination you want to jump to, i.e., the source that code generation was based on. The following code fragment illustrates a hyperlink and its possible destinations:

```
You can open the **Code Generator Options dialog** via the **All Options** button on the Advanced Page of the TargetLink Main Dialog.

The option settings are saved to a TargetLink model, so a TargetLink model must always be open if you want to specify option settings. The name of the TargetLink model to which the settings are saved is shown in the title bar of the dialog. Returning to the Advanced Page saves your settings only in the TargetLink Main Dialog. To save them at a TargetLink model, you must confirm your changes by pressing Apply or Close in the TargetLink Main Dialog.

For details, refer to [Code Generator Options Dialog](TargetLink Block and Object Reference).

**Related topics**

- Advanced Page (Main Dialog Block)
- tl_post_codegen_hook
- tl_pre_codegen_hook
Inheritance and Propagation of Signal Properties

For production code generation, TargetLink requires the specification of signal properties for each single block output. Until now, correct production code generation was possible only if you set the properties for each single block individually. Now the output variable of certain TargetLink blocks can inherit specific properties from the output variables of preceding TargetLink blocks:

- Data type
- Offset
- LSB
- Min
- Max
- Width

Inheritance and propagation of signal properties ensures maximum consistency of block properties in a model and reduces the amount of manual block property editing.

Basically, only blocks that do not perform any arithmetics can inherit preceding signal properties.

- Assignment block
- Bus Inport/Outport block that does not specify the interface of a root-level TargetLink subsystem. Each single signal of a bus has its own Inherit properties checkbox.
- D-Flip-Flop block (outport Q)
- D-Latch block (outport Q)
- Inport/Outport block that does not specify the interface of a root-level TargetLink subsystem.
- Merge block
- MinMax block
- Multiport Switch block
- Saturate block
New Features of TargetLink 2.2

- Switch block
- Unit Delay block
- Unit Delay Reset Enabled block

For details, refer to Basics on Inheritance of Signal Properties (TargetLink Production Code Generation Guide).

## Related topics

- Basics
  - Basics on Inheritance of Signal Properties (TargetLink Production Code Generation Guide)
  - Inheritance of Scaling Parameters

## Changes on Autoscaling

### Scaling reviewed flag

The Scaling reviewed option is now also available for Bool, Float and user-data types. It is reset when TargetLink blocks are copied.

For details, refer to Valid Scaling Parameters (TargetLink Production Code Generation Guide).

### Autoscaling also for Stateflow Inports and Outports

Stateflow Inports and Outports can hold scaling data which is used to scale surrounding TargetLink blocks. Thus you can avoid manual adaptation of interfaces between Simulink and Stateflow.
TargetLink Blocks Supporting Bus Signals

The following TargetLink blocks are now able to process bus signals:

- Merge block
- Multiport Switch block
- Switch block
- Unit Delay block

For details, refer to Basics of Buses in TargetLink (TargetLink Advanced Practices Guide).

Common TargetLink Context Menu Options

TargetLink offers the following options on the context menu of some edit fields in a block’s dialog (according to context):

<table>
<thead>
<tr>
<th>Button</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select</td>
<td>Lets you select a variable from the dSPACE Data Dictionary (DD).</td>
</tr>
<tr>
<td>Edit</td>
<td>Opens a DD dialog that lets you specify the DD variable’s properties.</td>
</tr>
<tr>
<td>Show in DD Manager</td>
<td>Opens the DD Manager and highlights the variable in the DD object tree (Data Dictionary Navigator).</td>
</tr>
<tr>
<td>Show Definition in Code</td>
<td>Opens the HTML code file related to the TargetLink subsystem the variable belongs to and highlights the variable’s definition.</td>
</tr>
<tr>
<td>Show Declaration in Code</td>
<td>Opens the HTML code file related to the TargetLink subsystem the variable belongs to and highlights the variable’s declaration.</td>
</tr>
<tr>
<td>Show Computation in Code</td>
<td>Opens the HTML code file related to the TargetLink subsystem the variable belongs to and highlights the variable’s computation.</td>
</tr>
</tbody>
</table>
Enhancements Regarding Stateflow

Easier user code integration in Stateflow
With the extern C Stateflow symbols feature, TargetLink allows calls of functions and the use of variables in Stateflow charts, which have not been defined in the Stateflow explorer, without the need to supply a script file for each customer-specific function. For details, refer to Extern C Stateflow Symbols (TargetLink Advanced Practices Guide).

Support for nested graphical functions
TargetLink supports graphical functions that are nested inside other graphical functions. They are treated like graphical functions that reside side by side. For details, refer to Basics on Graphical Functions (TargetLink Advanced Practices Guide).

Improved detection of if-then-else structures
A new flowchart transformation pattern for If-Then-Else-If flowcharts has been added. Even irregular-structured If-Then-Else-If flowcharts sharing some code for the individual conditional branches can now be transformed to well-structured nested if-then-else constructs. For details, refer to If-Then-Else-If Statement (TargetLink Advanced Practices Guide).

Miscellaneous Features

Access functions for struct components
TargetLink now supports access functions for struct components. Specifying access functions for struct components works similar to specifying access functions for scalars. For points you have to note when working with access functions for struct components, refer to Access Functions (TargetLink Advanced Practices Guide).

Better performance of MIL simulation
Up to now, the Switch block was implemented as an S-function. It is now basically implemented as a masked Simulink block. As a benefit of this, the performance of MIL simulations is improved by using an optimization in Simulink which ignores unused paths during simulation.
Transformation rule improvements

TargetLink uses transformations to represent multiplications in fixed point e.g. to perform scaling adjusts, i.e., to determine two natural numbers N (nominator) and D (denominator) that are applied if the input(s) and the output of a block must be given the same scaling (scaling adjust). Up to now, only the data type was considered for calculating the approximation parameters (N, D). The new algorithm considers both the data type of the involved operands and their ranges. The aim of this improvement is to optimize the efficiency of the generated code. In particular, 64-bit arithmetic can now often be avoided in cases where it was previously necessary. The generated production code can therefore differ at various places so that SIL and PIL simulations might be more accurate, or (within certain limits) worse than former TargetLink versions.

The following changes might occur in the generated production code:

- Changed constant values caused by the scaling adjust
- Modified data types or cast operations
- 64-bit macros can be either dropped or added (occurs only in rare cases if the accuracy of former approximation parameters (N, D) was not quite accurate)
- Change in the deviations between an MIL simulation and the production code simulations (SIL or PIL simulation mode)
- The deviations between MIL simulation and the production code simulations (SIL/PIL) can be greater (though they never exceed the quantization of the output)

For details, refer to Implementation of Integer Calculation Operations (TargetLink Production Code Generation Guide).

Support of the PolySpace Verifier

The integration of the PolySpace Verifier tool in TargetLink 2.2 allows you to distinguish between accidental overflows and CTOs used intentionally by TargetLink. For details, refer to ‘Compute-Through-Overflows’ Property of Two’s Complement Arithmetic (TargetLink Production Code Generation Guide).

ASAM-MCD 2MC enhancements

A2L File Export module

The A2L File Export module evaluates whether EPROMIdentifier and EPROMIdentifierAddress properties are available in the /<Application>/<Build>/TargetInfo dSPACE Data Dictionary node. If so, these properties are generated to the EPK and ADDR_EPK entries in the A2L file as follows:

- ADDR_EPK -> Property value of EPROMIdentifierAddress
- EPK -> Property value of EPROMIdentifier
New Features of TargetLink 2.2

For details, refer to Mapping of DD Objects to A2L File Elements (dSPACE Data Dictionary ASAM-MCD 2MC Import and Export).

**New import/export property** There is a new property DDSysytemVariablePath available for elements that are generated from variables in intermediate mode.

**Model conversion**
- The block property UserData is now preserved upon model conversion (SL2TL and TL2SL).
- The requirements info parameter, which was lost upon conversion from SL2TL and TL2SL, is now preserved.

**Long subsystem names** TargetLink now supports names of TargetLink subsystems with up to 58 characters (the previous limit was 26) for code generation.

**Storage of the active variant** The active variant is now preserved and saved with the DD file.

**TargetLink Blockset enhancements**
- Switching from full-featured (TargetLink Base Suite) to stand-alone (TargetLink Blockset) mode and vice versa is now possible without administrator rights.
- The TargetLink AutoDoc Customization block can now also be used in stand-alone mode.

**Discontinued TargetLink info file** TargetLink 2.2 is the latest version to generate a `<model>_info.m` file.

**Related topics**
- Basics
  - Multiplication by a Real Constant
  - Scaling Variables
  - Switch Block Description
  - Using Simulink Models in TargetLink

---

For details, refer to Mapping of DD Objects to A2L File Elements (dSPACE Data Dictionary ASAM-MCD 2MC Import and Export).
New Features of TargetLink 2.2

TargetLink New Features and Migration December 2006
New Features of dSPACE Data Dictionary 1.4

<table>
<thead>
<tr>
<th>Objective</th>
<th>dSPACE Data Dictionary 1.4 (DD) has the following new features, enhancements and changes:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New Key Features</strong></td>
<td></td>
</tr>
<tr>
<td><strong>General Changes in the dSPACE Data Dictionary</strong></td>
<td>35</td>
</tr>
<tr>
<td><strong>Changes in the Data Dictionary Manager</strong></td>
<td>36</td>
</tr>
<tr>
<td><strong>New and Modified DD API Commands</strong></td>
<td>39</td>
</tr>
<tr>
<td><strong>Changes in the ASAM-MCD 2MC Import and Export Module</strong></td>
<td>41</td>
</tr>
</tbody>
</table>

### New Key Features

<table>
<thead>
<tr>
<th>Objective</th>
<th>Information on the new key features of dSPACE Data Dictionary 1.4 is provided below.</th>
</tr>
</thead>
</table>

| Object Explorer pane | The Data Dictionary Manager provides a new pane to display and edit the properties of multiple DD objects. You can specify the DD objects to be shown in the same way as in the Property Manager. You can select single or multiple objects from the table, and edit them simultaneously by clicking the current value. If a limited set of values exists, you can select a new value from a list which opens; otherwise, |
you can enter a new value in a dialog which opens or an edit field. For further details, refer to Overview of the Data Dictionary Manager, Basics on Editing Multiple Data Dictionary Objects (dSPACE Data Dictionary Basic Concepts Guide) and Object Explorer (dSPACE Data Dictionary Manager Reference).
## General Changes in the dSPACE Data Dictionary

<table>
<thead>
<tr>
<th>Changes in the Autosave behavior of DD include files</th>
<th>In addition to switching the Autosave option on or off, you can also set the Autosave property of a DDIncludeFile object to specify that the DD include file is saved if objects in the included subtree were modified or that you are prompted if you want to save changes. For further details, refer to How to Specify Data Dictionary Include Files (dSPACE Data Dictionary Basic Concepts Guide) or DDIncludeFile (dSPACE Data Dictionary MATLAB API Reference).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifying the path of DD include files</td>
<td>In addition to specifying the path of a DD include file relative to the working folder, via a full path or via an environment variable, you can now also specify the path relative to the folder where the main DD project file resides. For further details, refer to How to Specify Data Dictionary Include Files (dSPACE Data Dictionary Basic Concepts Guide).</td>
</tr>
</tbody>
</table>
## Changes in the Data Dictionary Manager

### Changes in the Data Dictionary Manager

dSPACE Data Dictionary 1.4 includes the following enhancements:
- Object Explorer pane (refer to New Key Features on page 33)
- AUTOSAR SWC-D Merge Explorer
- DD Merge Explorer
- Import and Export of MATLAB data

### AUTOSAR SWC-D Merge Explorer

The AUTOSAR SWC-D Explorer displays the hierarchical structure of an AUTOSAR Software Component Description file compared to the currently open DD project file. It uses the same pane as the Model Browser or the DD Merge Explorer, i.e., only one of the tools can be shown at a time.

![AUTOSAR SWC-D Merge Explorer](image)

The reference file (AUTOSAR Software Component Description file) is shown in the AUTOSAR SWC-D Merge Explorer, where colored icons identify the differences. The context menu of the AUTOSAR SWC-D Merge Explorer gives you functions for copying objects to the currently open DD project file and generating a difference report. For further details, refer to Overview of the Data Dictionary Manager (dSPACE Data Dictionary Basic Concepts Guide).
DD Merge Explorer

The DD Merge Explorer displays the hierarchical structure of another DD project file compared to the currently open one. It uses the same pane as the Model Browser or the AUTOSAR SWC-D Merge Explorer, i.e., only one of the tools can be shown at a time.

The reference file (another DD project file) is shown in the DD Merge Explorer, where colored icons identify the differences. The context menu of the DD Merge Explorer gives you functions for copying objects to the currently open DD project file and generating a difference report. For further details, refer to Overview of the Data Dictionary Manager (dSPACE Data Dictionary Basic Concepts Guide).
New Features of dSPACE Data Dictionary 1.4

Import and export of MATLAB data

You can import variable objects from the MATLAB base workspace into the DD or export variable objects from the DD into MATLAB. In addition, you can import/export variable objects from/to MATLAB files, for example, .m and .mat. TargetLink also lets you synchronize MATLAB variable objects with the value properties of exported DD variable objects, and in the other direction, the value properties of existing DD variable objects with the values of imported MATLAB variable objects.

You can also import and export MATLAB data via API commands. For further details, refer to New and Modified DD API Commands on page 39.
New and Modified DD API Commands

New DD API commands

You can import variable objects from the MATLAB base workspace into the DD or export variable objects from the DD into MATLAB. In addition, you can import/export variable objects from/to MATLAB files (m. and mat.) by entering `dsdd_mlie_import ('file', '<filename>')` or `dsdd_mlie_export ('file', '<filename>')` in the MATLAB Command Window.

For easier input handling, the DD Manager offers one specialized import and one export dialog. The dialogs can be opened via the MATLAB Command Window by typing `dsdd_mlie_import_dlg` or `dsdd_mlie_export_dlg`.

For details, refer to How to Export Variable Objects via the Data Dictionary Manager (dSPACE Data Dictionary Basic Concepts Guide) and How to Import Variable Objects via the MATLAB Import Export API (dSPACE Data Dictionary Basic Concepts Guide).

Changed default with Upgrade attribute of Open command

The default value of the `dsdd('open', <filename>, 'Upgrade',<enumvalue>);` command has been changed to "ask". For details, refer to Open (dSPACE Data Dictionary MATLAB API Reference).

New modified component of object struct

The object struct, which is used to describe the attributes of a DD object in the dSPACE Data Dictionary MATLAB API, has a new component which signals that the respective object has been modified: `modified`. For details, refer to The Object Struct (dSPACE Data Dictionary MATLAB API Reference).

Modified API command `dsdd('SetAutoSave', ...)`

With previous TargetLink versions, the command `dsdd('SetAutoSave', '/Config/DDIncludeFiles/MyDDIncludeFile', 1)` set the AutoSave property to `on`. It is now set to `off`, because the AutoSave property is now an enumeration type. For details, refer to SetAutoSave (dSPACE Data Dictionary MATLAB API Reference).

Obsolete API command `dsdd('SetEnv', ..., 'ActiveVariant', ..., 'VariantName')`  

Up to now, `dsdd('SetEnv','ActiveVariant','VariantName')` and `dsdd('GetEnv','ActiveVariant')` allowed you to set and get the active VariantConfig object. Using the DD environment variable ActiveVariant to define the active VariantConfig object is now obsolete. Use the commands `dsdd('SetActiveVariant','VariantName')` and
dd('GetActiveVariant') instead, to which the old commands are redirected. GetActiveVariant also provides the handle of the active VariantConfig object. Alternatively, you can use the commands GetTargetLinkActiveVariantConfig and SetTargetLinkActiveVariantConfig. Both require that an Config/TargetLink object exist.

However, the SetEnv command will be supported as long as DD environment variables exist. For details, refer to GetActiveVariant and SetActiveVariant (dSPACE Data Dictionary MATLAB API Reference).

**Modified API command**

**AddDataVariantItem**

The syntax of the AddDataVariantItem command has changed from

```
errorCode = dsdd('AddDataVariantItem',<objectIdentifier>,<referenceValue>);
```

with dSPACE Data Dictionary 1.3 to

```
[hDDObject,errorCode] = dsdd('AddDataVariantItem',<objectIdentifier>);  
```

with dSPACE Data Dictionary 1.4. For details, refer to AddDataVariantItem (dSPACE Data Dictionary MATLAB API Reference).

**New API commands**

dSPACE Data Dictionary 1.4 provides the following new API commands:

- GetActiveVariant (dSPACE Data Dictionary MATLAB API Reference)
- GetDefaultPath (dSPACE Data Dictionary MATLAB API Reference)
- GetModified (dSPACE Data Dictionary MATLAB API Reference)
- HasObjectDialog (dSPACE Data Dictionary MATLAB API Reference)
- SetActiveVariant (dSPACE Data Dictionary MATLAB API Reference)
- SetModified (dSPACE Data Dictionary MATLAB API Reference)

**Obsolete API commands**

With dSPACE Data Dictionary 1.4, the following commands have become obsolete:

- GetDataVariantItem
- SetDataVariantItem
- GetDataVariantItemTarget
Changes in the ASAM-MCD 2MC Import and Export Module

| ASAP2 export supports ‘EPROMIdentifier’ and ‘EPROMIdentifierAddress’ | As of dSPACE Data Dictionary 1.4, the EPROMIdentifier and EPROMIdentifierAddress DD objects are supported. For further information, refer to Mapping of DD Objects to A2L File Elements (dSPACE Data Dictionary ASAM-MCD 2MC Import and Export). |
|----------------------------------------------------------------------------------|
| Maintaining the reference to variable objects during ASAP2 file generation | The reference to the DD variable object that the corresponding CHARACTERISTIC, MEASUREMENT, AXIS_PTS or AXIS_DESCR element has been generated for is maintained. It is saved as a DD path under the DDSubsystemVariablePath property. For further information, refer to Representation of Data During A2L File Export (dSPACE Data Dictionary ASAM-MCD 2MC Import and Export). |
# Migrating to TargetLink 2.2

When migrating from TargetLink Release 2.1 to TargetLink Release 2.2, you should be aware of the following aspects:

## Default Behavior of the Code Generator

Output variables of Merge blocks and DynamicSelector blocks (Selector blocks with an external source of element indices) can now be vectors or structures.

**Example**  
Suppose your model contains a Merge block with the Inherit signal properties checkbox selected. The Merge block is driven by signals with the same base types. In previous TargetLink versions, one vector output variable is generated for the Merge block. Since TargetLink 2.2, 1...n output variables are generated for the Merge block. The output variables have the structure of the output signal driving the first input of the Merge block. For example, if the Merge block is driven by a bus port block which maps the bus signal to structured variables of the `s1_tp` and `s2_tp` type, its output variables are also two structured variables of the `s1_tp` and `s2_tp` type.

## Where to go from here

Information in this section

| Default Behavior of the Code Generator | 43 |
| Simulation Behavior                   | 47 |
| Build Process                         | 47 |
Unspecified Simulink Inports and Outports

Several formal parameters can be generated for Simulink Inport and Outport blocks which are not connected to TargetLink InPort or OutPort blocks (unspecified Inports and Outports), depending on the blocks driving the Simulink Inport and Outport blocks.

**Example** In the following example model, the unspecified in_A inport and out_B outport are each mapped to a scalar and a vector. The variable class of the `VCT_SLFcnInput` variable class template is set to `FCN_ARG` in the DD.

TargetLink 2.2 generates the following function definitions and function calls:

```c
Void Sa2_sys_A(Int16 Sa2_in_A, Int16 Sa2_in_A_a[3])
{
    ...
}

Void Sa3_sys_B(Int16 * Sa3_out_B, Int16 Sa3_out_B_a[3])
{
    ...
}

Sa3_sys_B(&(Sa3_out_B), Sa3_out_B_a);
...
Sa2_sys_A(Sa1_Gain_X, Sa1_Gain_Y);
```

You can force TargetLink to map unspecified Simulink Inports and Outports to exactly one variable by connecting them to TargetLink InPorts and OutPorts.
Global interface variables

Suppose a so called data store net is mapped to global interface variables (IF variables). IF variables can now be vectors or structs.

**Example** In the following example model, Gain_SCAL, Gain_VEC and BusOutport_STRUCT write to IF variables. sys_A and sys_B read from the IF variables.

When generating production code, TargetLink maps the data store net to the following IF variables:

```
/******************************************************************************
SLGlobal: Default storage class for global variables | Width: 16
******************************************************************************/
Int16 IF_Sa1_Bus_Outport_STRUCT;
Int16 IF_Sa1_Bus_Outport_STRUCT_a;
Int16 IF_Sa1_Gain_SCAL;
Int16 IF_Sa1_Gain_VEC;
Int16 IF_Sa1_Gain_VEC_a;
Int16 IF_Sa1_Gain_VEC_b;
/******************************************************************************
SLGlobalInit: Default storage class for global variables with initial value | Width: 16
******************************************************************************/
Int16 Sa2_OutPort1_a[3] =
{ /*[0..2]*/ 0, 0, 0 /* 0., 0., 0. */
} ;
Int16 Sa2_OutPort2_a[2] =
{ /*[0..1]*/ 0, 0 /* 0., 0. */
} ;
Int16 Sa2_OutPort_a = 0;
Int16 Sa2_OutPort1_a[3] =
```
When Stateflow charts are used in reused subsystems, the names of sub-reuse structures and the sequence of their definitions can be different to previous TargetLink versions. This is caused by extensions and improvements to the code patterns for function reuse.

TargetLink now lets you modify or set additional code generation options via the Code Generator Options dialog. If you want to set code generation options via the Code Generator Options dialog for existing models, you have to delete code generation options in the relevant hook files that would otherwise override the options specified in the dialog.
Simulation Behavior

Modified MIL simulation behavior of the Switch block

Up to now, the Switch block was implemented as an S-function. It is now basically implemented as a masked Simulink block. An MIL simulation therefore behaves like a Simulink simulation, but the production code simulation (SIL/PIL simulation) does not necessarily do so. The difference occurs if the control input is a real integer signal, but the threshold value is a double. In this case, Simulink performs a type cast operation, i.e., the double is cast to integer and then compared, while in the production code, the comparison is implemented as expected.

Example

Suppose the input signal for the control input is a square wave between 0 and 1, and the threshold is 0.1.

Simulink casts the threshold to an integer of 0, performs a greater than or equal comparison between the input and the threshold value, and thus always calculates TRUE. With the generated production code, this comparison calculates different results: TRUE if the input signal is high (1), and FALSE if it is low (0).

As a consequence, the MIL simulation behavior can deviate from that of previous TargetLink versions, namely if the control input is an integer and the threshold is double. In addition, simulation differences between MIL and SIL/PIL simulations might occur, which was not the case with former TargetLink versions.

Build Process

Definition of extern global interface variables

In previous TargetLink versions, the definitions of extern global interface variables, if needed, were created in the simulation frame file, compiled and linked to the simulation application by default. This was done independently on the value of the ‘alias’ property of the corresponding variable class. In TargetLink 2.2, this is done only for extern global variables with the alias property set to off.
Migrating to dSPACE Data Dictionary 1.4

When migrating from dSPACE Data Dictionary 1.1, 1.2 or 1.3 to 1.4, you should be aware of the following aspects:

<table>
<thead>
<tr>
<th>Where to go from here</th>
<th>Information in this section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrading DD project files</td>
<td>Upgrading Data Dictionary Project Files 50</td>
</tr>
<tr>
<td></td>
<td>How to Upgrade DD Project Files via the Data Dictionary Manager 52</td>
</tr>
<tr>
<td></td>
<td>Changed or Obsolete DD API Commands 53</td>
</tr>
<tr>
<td></td>
<td>Adapting the Target_Info file for A2L File Export 54</td>
</tr>
</tbody>
</table>
Upgrading Data Dictionary Project Files

Changes in the data model

Changes have been made to the data model of the dSPACE Data Dictionary. For example, some objects have additional properties or child objects, and some object properties have been left out. You therefore have to upgrade DD project files you created with dSPACE Data Dictionary 1.1, 1.2 or 1.3. The dSPACE Data Dictionary provides an upgrade tool for this. If you invoke the upgrade tool, it checks the DataModelRevision property in the DD/Config/General subnode.

If the value of this property is less than 132, the DD project file must be upgraded.

If you upgrade your DD project file, the following problems can occur:

- If you have insufficient access rights, the DD project file might not be upgraded. In this case you should change to the admin mode via the Extras menu of the Data Dictionary Manager.

Repeat the DD project file upgrade afterwards using the Upgrade Current DD File option from the Tools menu of the Data Dictionary Manager.
Changes of objects which reside in DD files to be included are not saved automatically. To prevent the changes from being lost, open the DD Manager and set the AutoSave property for the DD include files to 'PromptIfModified'.

Save the DD project file afterwards.

After the upgrade, set the AutoSave property back to 'off', if necessary.

For more information, refer to How to Upgrade DD Project Files via the Data Dictionary Manager on page 52.
How to Upgrade DD Project Files via the Data Dictionary Manager

Objective
You can upgrade the a DD project file by opening it via the Data Dictionary Manager.

Method
To upgrade DD project files via the Data Dictionary Manager
1. From the File menu in the Data Dictionary Manager, select Open DD Project File.
2. In the Open dialog, select the relevant DD project file and click Open. You are prompted to specify whether the DD project file should be upgraded or not.

3. Click Yes.

Result
The selected DD project file is opened and upgraded.

Alternatively, you can choose one of the following ways of upgrading a DD project file:
- Load a DD project file and enter
  ```matlab
erroCode = dsdd('Upgrade');
```
in the MATLAB Command Window.
- In the MATLAB Command Window, enter
  ```matlab
erroCode = dsdd('Open','file',<DDFile>,Upgrade,<upgrade>);
```

For details, refer to Open and Upgrade ( dSPACE Data Dictionary MATLAB API Reference).
The following table shows the parameters and their possible values:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDFile</td>
<td>Name of DD file to open (string)</td>
</tr>
<tr>
<td>upgrade</td>
<td>&quot;on&quot;, &quot;off&quot;, or &quot;ask&quot; (default = 'ask')</td>
</tr>
</tbody>
</table>

If upgrading the DD project file has failed, you can open the Message Browser to show error messages in the Message Browser using the following command:

```matlab
if errorCode ~= 0
d_error_register(dsdd('GetMessageList'));
d_msgdlg;
end
```

Related topics
- Open DD Project File (dSPACE Data Dictionary Manager Reference)

## Changed or Obsolete DD API Commands

**Adapting your scripts to changed DD API commands**

Since several DD API commands have been modified or are obsolete, you may have to adapt your scripts accordingly. For details on changes on the DD API commands, refer to New and Modified DD API Commands on page 39.
Adapting the Target_Info file for A2L File Export

As of dSPACE Data Dictionary 1.4, the edit field to specify the compiler that the ECU application has been build with was removed from the Manage Build dialog.

The compiler information is now read from the target_info.m file. If you use the target_info.m files delivered with the TargetLink Target Simulation Module, the information about the used compiler is already contained there. If you use your own target_info.m file created for DD 1.3.x you have to add the targetInfo.cc component to the targetInfo structure defined in the file.

For further information, refer to Providing a Target Info File (dSPACE Data Dictionary ASAM-MCD 2MC Import and Export).
Index

A
AUTOSAR SWC support
  TargetLink AUTOSAR blockset 18

M
MATLAB
  TargetLink compatibility 15

T
TargetLink
  Java Virtual Machine compatibility 13
  MATLAB compatibility 15