TargetLink

Blockset Guide

For TargetLink 5.1

Release 2020-B – November 2020

dSPACE

How to Contact dSPACE

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

How to Contact dSPACE Support

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

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

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

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

Software Updates and Patches

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

Important Notice

This publication contains proprietary information that is protected by copyright. All rights are reserved. The publication may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the publication 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.

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

AUTERA, ConfigurationDesk, ControlDesk, MicroAutoBox, MicroLabBox, 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.

The ability of dSPACE TargetLink to generate C code from certain MATLAB code in Simulink®/Stateflow® models is provided subject to a license granted to dSPACE by The MathWorks, Inc. MATLAB, Simulink, and Stateflow are trademarks or registered trademarks of The MathWorks, Inc. in the United States of America or in other countries or both.

Contents

About This Guide

Contents

This guide explains how to use the TargetLink Modeling Only operation mode. TargetLink lets you switch between two operation modes:

- Modeling Only operation mode, used for designing a model.
- Full-Featured operation mode, used for generating code.

Developers, using these two operation modes, can work on a model and can exchange it among each other in the course of development.

Symbols

dSPACE user documentation uses the following symbols:

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

Naming conventions

dSPACE user documentation uses the following naming conventions:

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

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

Special folders

Some software products use the following special folders:

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

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

or

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

Documents folder A standard folder for user-specific documents.

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

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

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

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as Adobe® PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

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

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

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

Introduction to the TargetLink Operation Modes

Overview of the TargetLink Operation Modes

TargetLink operation modes

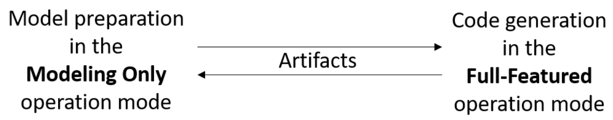
TargetLink lets you switch between the following operation modes:

- Modeling Only
- Full-Featured

The Modeling Only operation mode is intended for function developers. Function developers usually prepare and design a model without generating production code for it. In the course of development, this model is exchanged with software developers using the Full-Featured operation mode. Full-Featured users are able to generate production code for the model.

Model-based development means that function developers design control algorithms from which software developers generate C code with TargetLink, which is then executed on ECUs. Both groups of developers work on the same model but use different operation modes.

Compared to the Full-Featured operation mode, the Modeling Only operation mode is not license-protected.



TargetLink features

Modeling Only	Full-Featured
▪ Model exchange in a workgroup (Simulink and TargetLink users) ▪ Inspecting block properties related to production code generation ▪ Simulating in double-precision with Simulink (MIL) ▪ Real-time simulation on dSPACE prototyping hardware ▪ Configuration of HTML document generation	▪ Production code generation ▪ Production code simulation on a PC, i.e., software-in-the-loop simulation (SIL) ▪ Production code simulation on a target, i.e., processor-in-the-loop simulation (PIL) ▪ Data logging in all modes, letting you compare data ▪ Autoscaling of variables ▪ Generation of HTML documents ▪ Export of relevant files of a software project ▪ Property Manager ▪ Preparation of Simulink systems for TargetLink (model preparation)

Operation mode switch

You can switch between the Modeling Only operation mode and the Full-Featured operation mode via the following API function:

```
tlOperationMode('set', 'ModelingOnly')
```

Model exchange

Using both operation modes lets you link to crucial development phases:

1. Prototyping using Simulink Coder
2. Production code generation using TargetLink

Function developers using the Modeling Only operation mode and Simulink Coder, and software developers using TargetLink, can exchange models without model preparation. This is useful because the developers frequently exchange models in the course of development. For more information on exchanging models, refer to How to Use Models from Software Developers on page 24.

Some exchange processes can modify model element paths so that it is necessary to remove simulation frames. For more information, refer to How to Remove TargetLink Simulation Frames (API command) on page 16.

Data Dictionary

TargetLink uses the Data Dictionary (DD) as a central data container that holds all information relevant for code generation.

It stores important variables, such as calibratable parameters and measured variables. If you want to reference variables, you can use TargetLink block dialogs. For more information, refer to TargetLink Block Dialogs on page 16.

To edit the DD, dSPACE provides the DD Manager, which requires the Full-Featured operation mode.

The default DD is only a template DD that is initially connected. Models that you receive from software developers must be connected to a project-specific DD.

Simulation

The Modeling Only operation mode is compatible with Simulink® Coder^{TM}, Real-Time Interface, and ConfigurationDesk/SCALEXIO.

The following types of simulation are possible:

- Simulink simulation is always carried out on the Simulink platform and not in real time.
- Real-time simulation proceeds at a rate that matches that of the real process it simulates.

For more information, refer to Simulation Basics on page 19.

Related topics

References

tlOperationMode (📖 TargetLink API Reference)

Creating Models in the Modeling Only Operation Mode

Approaches

If you are familiar with MATLAB®/Simulink®, you basically have the knowledge to design a TargetLink model, too. You can create a model in two ways:

- From scratch
- By modifying an existing model

Where to go from here

Information in this section

Supported Blocks

Overview

To implement your control algorithm with TargetLink, you can use the following blocks in your TargetLink subsystem:

- TargetLink simulation blocks (including blocks for the implementation of a real-time operating system)
- TargetLink utility blocks
- Supported Simulink blocks

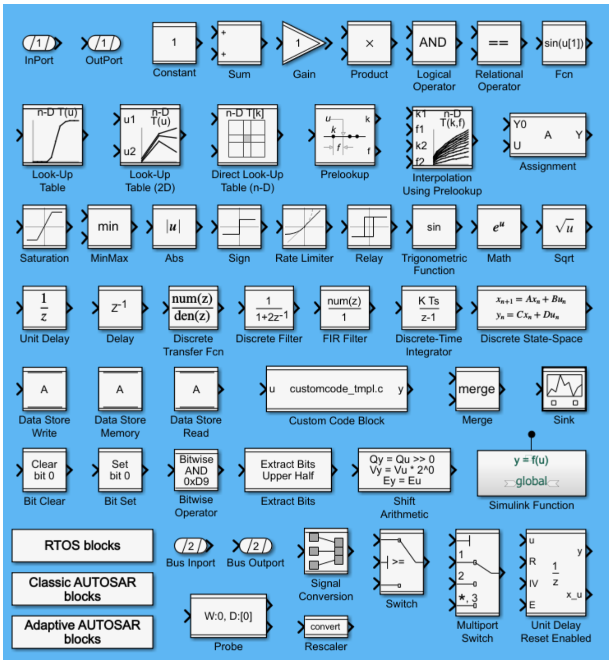
- Stateflow charts

> **Note**
>
> The Stateflow block library is accessible via the `sflib` command in the MATLAB Command Window.

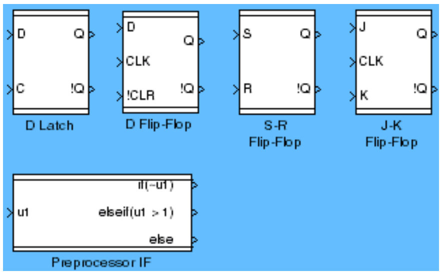
All the TargetLink blocks and the supported Simulink blocks are accessible via the TargetLink block library (tllib).

TargetLink simulation blocks

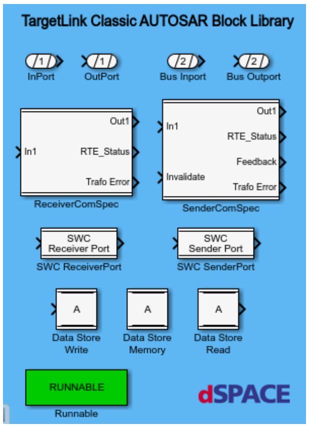
TargetLink simulation blocks can be found on the topmost level of the TargetLink block library (tllib).

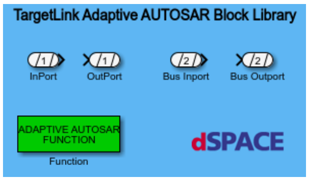


In the Extras sublibrary, the following blocks are available:



In the Classic AUTOSAR blocks and Adaptive AUTOSAR blocks sublibrary, blocks for modeling and generating code for AUTOSAR-compliant software components are available:



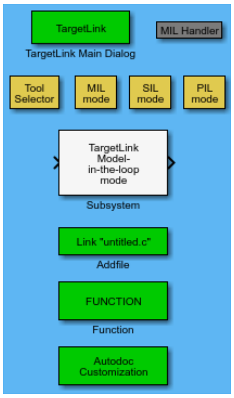

TargetLink Adaptive AUTOSAR Block Library
1
InPort
1
OutPort
2
Bus Inport
2
Bus Outport
ADAPTIVE AUTOSAR FUNCTION
Function
dSPACE

Note

The grayed block is not accessible in the Modeling Only operation mode.

TargetLink utility blocks

TargetLink utility blocks can be found on the topmost level of the TargetLink block library (tllib).

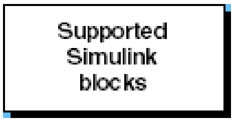


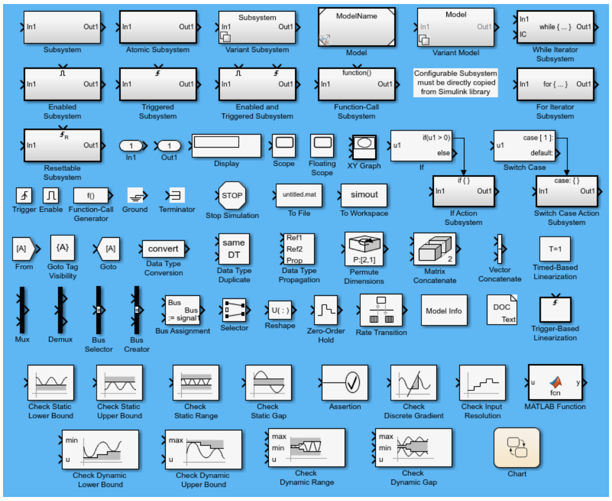
Note

The grayed blocks are not accessible in the Modeling Only operation mode.

Supported Simulink blocks

Simulink blocks supported for production code generation are located in the Supported Simulink blocks sublibrary:





How to Create a Model From Scratch

Objective

To create a model from scratch, you have to perform preparatory steps before you start modeling a control algorithm with TargetLink blocks.

Preconditions

To reference predefined DD objects in any block dialog, you need a project-specific DD providing all relevant objects, e.g., DD Variable objects. If no DD is associated with your model, TargetLink asks you to either associate an existing DD (project-specific DD provided by an owner of a Full-Featured license) or to create a new default DD that is based on templates. In this case, no project-specific DD objects are available. To correctly reference any predefined DD object, you have to know which objects are contained and how they are named. Ask your DD provider for a list. For more information, refer to How to Reference Values from DD Variable Objects to Block Variables on page 17.

Method

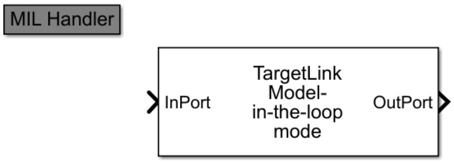
To create a model from scratch

1 Create a new Simulink model.

2 From the TargetLink menu, select Block Library.

3 From the TargetLink Block Library, drag a TargetLink Subsystem block and a MIL Handler block to the rootlevel of the model.

A simple TargetLink model



Note

TargetLink subsystems provide the TargetLink simulation frame, which lets you switch the simulation mode (MIL, SIL, or PIL). Working with the Modeling Only operation mode requires TargetLink simulation frames to be removed from TargetLink subsystems.

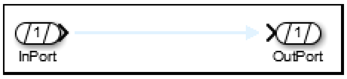
4 Click Subsystem and enter a name for the TargetLink subsystem.

Note

TargetLink subsystem names must follow conventions for MATLAB identifiers, which can only consist of digits, letters, and underscores. The first character must not be a digit or an underscore. With this version of TargetLink, the maximum number of characters for TargetLink subsystem names is 58.

5 Double-click the Subsystem block.

The corresponding TargetLink subsystem opens.



Result

In the TargetLink subsystem, you can model the control algorithm. Refer to Supported Blocks on page 11 to get an overview of which blocks are available for building a TargetLink subsystem.

Related topics

HowTos

How to Remove TargetLink Simulation Frames (API command).. 16

How to Remove TargetLink Simulation Frames (API command)

Objective

To remove Simulation frames from TargetLink subsystems.

TargetLink Simulation Frames

If you work in the Modeling Only operation mode and you receive a model from a Full-Featured user, it could be possible that TargetLink simulation frames have to be removed from TargetLink subsystems. This can be the case if a model element path changes. If no model element path changes, you do not have to remove simulation frames.

Method

To remove a TargetLink simulation frame

1 Load or open your model.

```
open_system('MyModel');
```

2 Clear `MySubsystem` in `MyModel`.

```
tl_clear_system('system',gcb,'removeSimFrameOnly','on',...
'remapOutputScalingData','off','remapSaturationFlags', 'off',...
'remapConstrainedLimits','off','remapSFObjectScalingData','off',
...
'logFileName','');
```

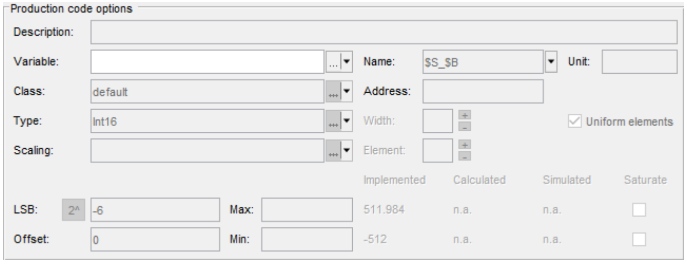
Result

The TargetLink simulation frame is removed from the selected TargetLink subsystem.

TargetLink Block Dialogs

TargetLink block dialogs

The dialog of a TargetLink block shows properties that are related to production code generation. Most of them are displayed in the **Production code options** group box of the dialog of a TargetLink block.



Production code options

Description:

Variable:

Name: $S_$B

Unit:

Class: default

Address:

Type: Int16

Width:

☑ Uniform elements

Scaling:

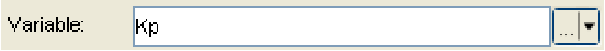
Element:

	Implemented	Calculated	Simulated	Saturate
LSB: 2^ -6 / Max:	511.984	n.a.	n.a.	□
Offset: 0 / Min:	-512	n.a.	n.a.	□

In the Modeling Only operation mode, all the Production code options are disabled, except for the **Variable** option.

DD Variable objects

To reference a DD Variable object, you have to open the block dialog. In the **Variable** edit field, you can enter the name of the variable or use the Browse button.


Variable: Kp

For this action, the appropriate DD project file `(*.dd)` must be active and must contain the desired variable.

TargetLink and Simulink dialogs

TargetLink provides an option that controls whether the TargetLink (default) or Simulink block dialog opens when you double-click a TargetLink block. You can use this option if you work in the Modeling Only operation mode and prefer to view and set block properties in Simulink dialogs.

Switch to Simulink block dialogs.

```
tl_pref('set','DialogProvider','Simulink')
```

Switch to TargetLink block dialogs.

```
tl_pref('set','DialogProvider','TargetLink')
```

How to Reference Values from DD Variable Objects to Block Variables

Objective

To reference values from DD **Variable** objects to block variables, for example, parameters and initial values.

This facilitates the following:

- Consistency between model and DD
- Variant coding

Precondition

You have referenced a DD **Variable** object to the block variable. Refer to How to Reference DD Variable Objects to Block Variables (📖 TargetLink Preparation and Simulation Guide).

Possible methods

You can reference values from DD **Variable** objects to block variables as follows:

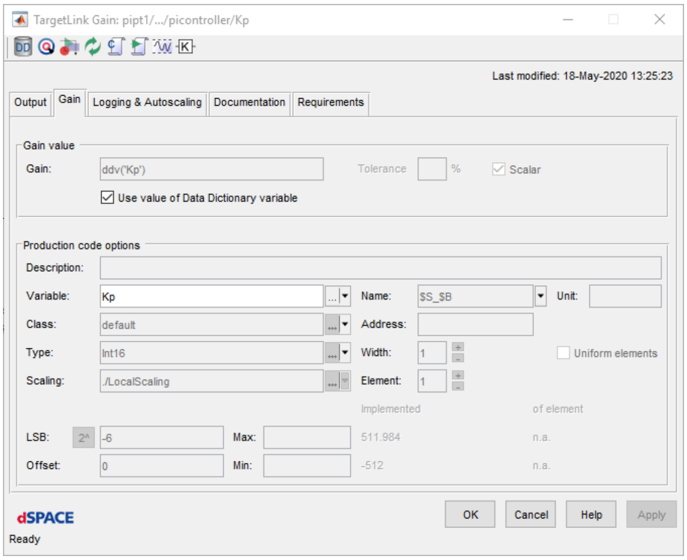
- Via block dialog. Refer to Method 1 on page 18.
- Via API. Refer to Method 2 on page 18.

Method 1

To reference values from DD Variable objects to block variables via block dialog

1 In the block dialog, select the Use value of Data Dictionary variable checkbox.

TargetLink automatically inserts `ddv(<VariableObject>)` in the value field.



TargetLink Gain: pipt1/.../picontroller/Kp

Last modified: 18-May-2020 13:25:23

Gain value

Gain: ddv('Kp')   Tolerance: %   ☑ Scalar

☑ Use value of Data Dictionary variable

Production code options

Description:

Variable: Kp   Name: $S_$B   Unit:

Class: default   Address:

Type: Int16   Width: 1   □ Uniform elements

Scaling: ./LocalScaling   Element: 1

				Implemented	of element
LSB:	2^ -6	Max:		511.984	n.a.
Offset:	0	Min:		-512	n.a.

Ready

Method 2

To reference values from DD Variable objects to block variables via API

1 Use the **`tl_set`** and **`ddv`** command to reference a value from a DD Variable object to a block variable.

```
tl_set(<Block>, <VariableProperty>, 'ddv(<VariableObject>)';
```

For example:

```
tl_set(<Block>, 'gain.value', 'ddv('allVars/myVar')';
```

Result

You have referenced a value from a DD Variable object to a block variable.

Related topics

HowTos

How to Reference DD Variable Objects to Block Variables (📖 TargetLink Preparation and Simulation Guide)

References

ddv (📖 TargetLink API Reference)
tl_set (📖 TargetLink API Reference)

Simulating Models

Simulink

The simulation behavior of TargetLink is the same as the simulation behavior of the Simulink.

Where to go from here

Information in this section

Simulation Basics

Compatibility with Simulink Coder, Real-Time Interface and ConfigurationDesk/SCALEXIO

The Modeling Only operation mode is compatible with Simulink® Coder^{TM}, Real-Time Interface and ConfigrationDesk/SCALEXIO. The options for simulating a model are the same as with Simulink. You can perform Simulink simulation and rapid control prototyping, both with floating-point precision.

The following types of simulation are possible:

- Simulink simulation
- Real-time simulation

Simulink simulation

Simulating a model offline means that you can perform fundamental behavior validity checks to support function development. The major feature of this type of simulation is that the calculations are performed under non-real-time conditions. The computer has as much time as needed to calculate the behavior of the simulated system.

Real-time simulation

Once a model has been validated by performing Simulink simulation, you may want to test the controller in combination with the actual plant. Since the designed controller might need further modifications, you do not want to produce it in hardware at this stage of development. You therefore have to connect the real plant to a controller that is simulated in real time. This technique is called rapid control prototyping (RCP).

You can implement a TargetLink controller model on dSPACE prototyping systems. This hardware provides sufficient computing power to calculate controller models in real time as well as an I/O interface to the plant that can be configured directly from the model. This convenient method of implementation is made possible by the blockset of Real-Time Interface or a combination between ConfigurationDesk (software) and SCALEXIO (hardware). These software packages establish communication between a simulated control system and the real plant. Real-Time Interface comes with documentation that shows in detail how to carry out the various steps of implementing dSPACE components in each of your TargetLink models. Refer to the *RTI and RTI-MP Implementation Guide*. For more information on ConfigurationDesk and SCALEXIO, refer to the respective documentation.

Simulink Coder

To simulate a model in real time, you need to generate real-time code from it. To generate real-time code from a model that has been designed using TargetLink, you can use Simulink Coder.

Evaluation

Evaluation of a simulation depends on the simulation type:

Simulink Simulation	Real-Time Simulation
For an overview of signals you can use Simulink scope blocks.	To apply sophisticated experimentation features, you can use ControlDesk, which allows you to perform tasks such as parameter studies.

Note

Sophisticated plot windows are available in TargetLink for production code generation only.

How to Generate an S-Function DLL File for Custom Code (Type I) Blocks

Objective

To use Custom Code (Type I) blocks compatible with the Modeling Only operation mode, you have to generate an S-function DLL file.

Custom Code (Type I) blocks

S-function DLL files of Custom Code (Type I) blocks must be compatible with your Modeling Only operation mode environment. In this operation mode, you can rebuild the S-function DLL files of a model taken from the TargetLink Base Suite, if necessary.

If you changed the Custom Code (Type I) block code or do not have the S-function DLL files, you have to generate the files.

DLL and TLC files

When you generate an S-function DLL file for a Custom Code (Type I) block and Simulink® Coder™ is installed, a TLC file is generated automatically. The TLC file enables the custom code to be inlined into the generated real-time code.

Simulink simulation

To simulate a Custom Code (Type I) block offline, you need only the related S-function DLL file.

Preconditions

For S-function DLL generation, the underlying custom code C file is required.

Possible methods

TargetLink lets you generate an S-function DLL for a Custom Code (Type I) block in two ways:

- Via block dialog
- Via API function

Method 1

To generate an S-function DLL for a Custom Code (Type I) block via dialog

1. In the dialog of the block, select the Code & Logging page.
2. In the File name edit field, enter the name of the custom code C file.
3. Click the Build S-function button.

Method 2

To generate an S-function DLL file for a Custom Code (Type I) block via API function

1. Specify the name of the custom code C-file: Type `tl_set(<CustomCodeBlockPath>,'codefile',<codeFileName>)` in the MATLAB Command Window.
2. Type `tl_build_customcode_sfcn (’Block’,<CustomCodeBlockPath>)` in the MATLAB Command Window.

Result

An S-function DLL file is generated from the underlying custom code C file.

> **Tip**
>
> Instead of specifying the Simulink path, you can select a block in the model and apply the `gcb` command, for example, `tl_set(gcb,'codefile',<codeFileName>)`.

Related topics

Basics

Custom Code Blocks (📖 TargetLink Preparation and Simulation Guide)

Exchanging Models

Introduction

You can easily transfer a model in a workgroup where one developer performs rapid control prototyping in the Modeling Only operation mode and another generates production code in the Full-Featured operation mode.

Where to go from here

Information in this section

Utilizing Base Suite Models

Introduction

You can perform rapid prototyping with a model that comes to you from the production code team.

How to Use Models from Software Developers

Objective

To use models from software developers as a Modeling Only user, you have to use all relevant files.

Relevant files

If you want to work on a model from a Full-Featured operation mode user, you have to receive the model file (`*.slx`), the Data Dictionary project file (`*.dd`), and all other relevant files.

The following files can be relevant:

- Scripts that generate workspace variables (`*.m`)
- Simulink DD (`*.sldd`)
- Custom code files (`*.c, *.h, *.tlc, _flp.mexw64`)

Simulation mode

The simulation mode has to be MIL.

Custom Code (Type I) blocks

If the model contains Custom Code blocks, you need their underlying C files, S-function DLL files, and TLC files, depending on the type of simulation. For more information, refer to How to Generate an S-Function DLL File for Custom Code (Type I) Blocks on page 20.

Method

To use a model from the production code team

1. Make sure that you have the Data Dictionary project file (`*.dd`).
2. If the model contains Custom Code blocks, make sure you have their template files, S-function DLL files, and TLC files.
3. If the model contains user-defined blocks, make sure that you have all the library files and libmap files.
4. Make sure that you have the model file (`*.mdl` or `*.slx`).

Result

You can now use the model.

Related topics

HowTos

How to Generate an S-Function DLL File for Custom Code (Type I) Blocks 20

Examples

Example of Working in the Modeling Only operation mode

Introduction

A typical scenario is that production code generation and rapid control prototyping are performed iteratively. It should not be necessary to convert the TargetLink model.

Workflow

Suppose you work in the Modeling Only operation mode. You may want to perform rapid control prototyping with a TargetLink model that has already been prepared for production code generation.

The process you are involved in could be organized as follows:

1. A colleague engaged in production code generation hands out the model file, all files related to the model, and the DD project file (TargetLink Data Dictionary).
2. You just open the model file in the Modeling Only operation mode.
3. You may insert new TargetLink blocks into the controller model.
4. You remove all blocks that represent or belong to the simulated plant. You replace them by **Real-Time Interface** blocks to establish the I/O interface to the real plant.
5. If the model contains Custom Code blocks, you generate S-function DLL and TLC files from the custom code C files.
6. You generate real-time code For the TargetLink controller model using RTI Blockset, DSMPBLib or TargetLink.
7. You simulate the controller in real time on dSPACE prototyping hardware and perform parameter studies using ControlDesk.
8. When the model meets your requirements, you remove all **Real-Time Interface** blocks and replace them by blocks representing the plant.
9. You give the modified model file back to your colleague.

Related topics

References

tlOperationMode (🕮 TargetLink API Reference)