

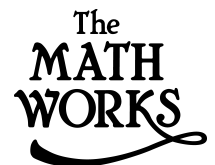
# SB2SL

## Model-Based and System-Based Design

Modeling

Simulation

Implementation



User's Guide

*Version 2*

## SB2SL

### A SystemBuild™ to Simulink® Translator User's Guide

Xmath is a registered trademark and SystemBuild is a trademark of Wind River Systems, Inc. with which The MathWorks, Inc. is not affiliated.

#### How to Contact The MathWorks:



www.mathworks.com	Web
comp.soft-sys.matlab	Newsgroup



support@mathworks.com	Technical support
suggest@mathworks.com	Product enhancement suggestions
bugs@mathworks.com	Bug reports
doc@mathworks.com	Documentation error reports
service@mathworks.com	Order status, license renewals, passcodes
info@mathworks.com	Sales, pricing, and general information



508-647-7000	Phone
--------------	-------



508-647-7001	Fax
--------------	-----



The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098	Mail
--------------------------------------------------------------------	------

For contact information about worldwide offices, see the MathWorks Web site.

© COPYRIGHT 1998 - 2004 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

**FEDERAL ACQUISITION:** This provision applies to all acquisitions of the Program and Documentation by or for the federal government of the United States. By accepting delivery of the Program, the government hereby agrees that this software qualifies as "commercial" computer software within the meaning of FAR Part 12.212, DFARS Part 227.7202-1, DFARS Part 227.7202-3, DFARS Part 252.227-7013, and DFARS Part 252.227-7014. The terms and conditions of The MathWorks, Inc. Software License Agreement shall pertain to the government's use and disclosure of the Program and Documentation, and shall supersede any conflicting contractual terms or conditions. If this license fails to meet the government's minimum needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to MathWorks.

MATLAB, Simulink, Stateflow, Handle Graphics, and Real-Time Workshop are registered trademarks, and Target Language Compiler is a trademark of The MathWorks, Inc.

Other product or brand names are trademarks or registered trademarks of their respective holders.

Printing History:	October 1998	Online only	Revised for Version 2 (Release 11)
	September 1999	Online only	Minor revisions for Release 11.1
	September 2000	Online only	Minor revisions for Release 12
	June 2001	Online only	Minor revisions for Release 12.1
	October 2004	Online only	Minor revisions for Release 13SP2

## SB2SL

---

<b>Introduction</b> .....	<b>2</b>
Software Requirements .....	2
Installation .....	3
<b>Using SB2SL</b> .....	<b>4</b>
Loading a SystemBuild Model into SB2SL .....	5
Selecting SystemBuild SuperBlocks .....	6
Selecting a SuperBlock Partition for Conversion .....	7
Converting SuperBlocks to Simulink Models .....	8
Compiling Converted BlockScript .....	9
Saving Translated Models and Data .....	10
Generating a Report .....	11
Setting Translation Options .....	11
<b>Compatibility Between SystemBuild and Simulink</b> .....	<b>15</b>
SB2SL Simulink Library .....	15
Using Real-Time Workshop with Converted SB2SL Models ...	16
<b>Limitations</b> .....	<b>17</b>
File Formats .....	17
Blocks Not Converted to Simulink .....	17



# SB2SL

---

<b>Introduction</b> . . . . .	2
Software Requirements . . . . .	2
Installation . . . . .	3
<b>Using SB2SL</b> . . . . .	4
Loading a SystemBuild Model into SB2SL . . . . .	5
Selecting SystemBuild SuperBlocks . . . . .	6
Selecting a SuperBlock Partition for Conversion . . . . .	7
Converting SuperBlocks to Simulink Models . . . . .	8
Compiling Converted BlockScript . . . . .	9
Saving Translated Models and Data . . . . .	10
Generating a Report . . . . .	11
Setting Translation Options . . . . .	11
<b>Compatibility Between SystemBuild and Simulink</b> . . . . .	15
SB2SL Simulink Library . . . . .	15
Using Real-Time Workshop with Converted SB2SL Models . . . . .	16
<b>Limitations</b> . . . . .	17
File Formats . . . . .	17
Blocks Not Converted to Simulink . . . . .	17

## Introduction

You can translate SystemBuild™ SuperBlocks to Simulink® models using SB2SL. For each SystemBuild SuperBlock in your model, you can:

- Create a Simulink model that represents the structure and hierarchy of your SystemBuild model
- Translate Xmath® data from the SystemBuild model into MATLAB® variables in the MATLAB workspace
- Produce a report providing details of the translation

SB2SL translation is performed on a block-by-block basis. Except for a few blocks, all SystemBuild blocks are translated into either:

- Its Simulink counterpart, or
- A masked subsystem block containing the computational equivalent if no Simulink counterpart exists

A few blocks are not translated, but appropriate blank placeholder blocks are created in the resulting Simulink model.

Once you translate your SystemBuild model into Simulink, the results of the Simulink simulation match the results of a SystemBuild simulation. However, due to modeling differences between the two environments, you may want to perform further model optimizations in order to achieve top simulation performance. We strongly recommend that you validate all models after translation.

## Software Requirements

Version 2.1 of SB2SL requires MATLAB 6 and Simulink 4. For general system requirements, see the *Installation Guide* for your platform.

You can apply SB2SL to SystemBuild files saved from Version 5.0 through Version 6.2 of SystemBuild UNIX or PC in ASCII format. However, new blocks introduced since SystemBuild Version 6.0 cannot be converted. For more information, see the list of blocks not converted in the “Limitations” section.

## **Installation**

See the *Installation Guide* for information on installing MATLAB and Simulink on your platform.

## Using SB2SL

Before translating a SystemBuild model, you must save it in ASCII format (usually with a file extension `.xmd`). SB2SL does not support translation of binary SystemBuild files.

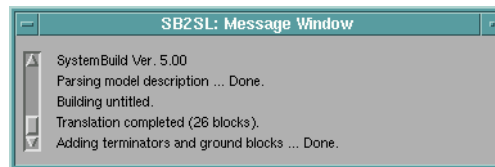
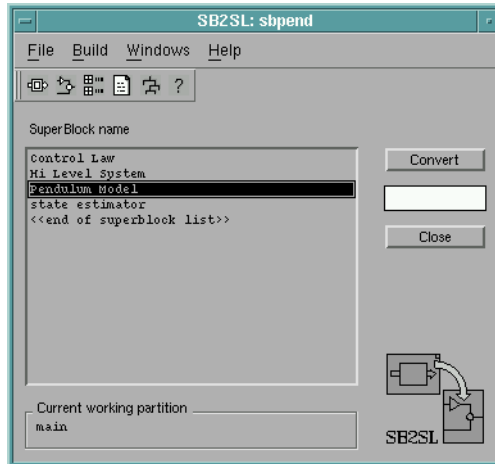
To make possible the transfer of parameterized variables (`%vars`) from SystemBuild to Simulink you must also make sure the variables are declared and resident in the Xmath workspace. Then save the SystemBuild model with the **Xmath Variables** option toggled to **Save All**.

To start SB2SL, at the MATLAB command prompt type

```
sb2sl
```



This opens the main SB2SL graphical user interface (GUI) and an associated message window.



**Figure 1: The Main SB2SL GUI and the Message Window**

## Loading a SystemBuild Model into SB2SL

Choose **Open** from the **File** menu in the main window to load a SystemBuild model. This opens a file browser from which you can select a SystemBuild model file. Once you select the name of a SystemBuild file in the browser, SB2SL:

- Opens the file
- Loads all of the parameters, if any, into the MATLAB workspace
- Lists the names of all the SuperBlocks in your model in a listbox

You can follow along with this tutorial by loading the `.xmd` file, `sbspend.xmd`, included with SB2SL.

---

**Hint:** To locate the directory from which to browse for `sbspend.xmd`, type

---

```
which sbspend.xmd
```

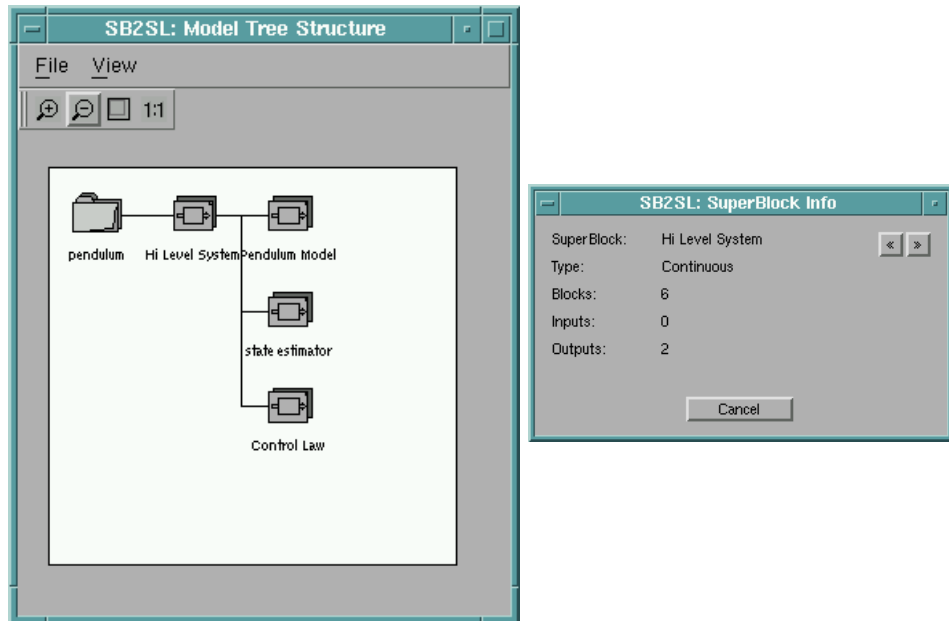
at the MATLAB command line.

## Selecting SystemBuild SuperBlocks

You can use SB2SL to convert SystemBuild SuperBlocks to Simulink models at any level in the SystemBuild hierarchy. To begin the process of Superblock conversion, select the name of a top level SuperBlock you want to convert from the list in the main SB2SL GUI. This action highlights all SuperBlock names referenced by the selected SuperBlock.

Alternatively, you can display the SuperBlocks in a tree-like file structure by selecting **Tree** from the **Windows** menu on the main GUI. This opens the **Model Tree Structure** window. From this window, you can use your mouse to select the SuperBlock you want to convert to a Simulink diagram.

If you right-click on a SuperBlock icon, a window opens that contains additional information related to that SuperBlock (type, number of blocks, etc.).

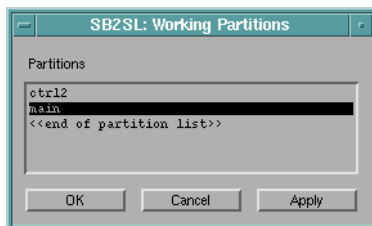


**Figure 2: Model Tree Structure and SuperBlock Information Windows**

## Selecting a SuperBlock Partition for Conversion

A SystemBuild model can contain data in separate partitions associated with each SuperBlock. When you load a SystemBuild model into SB2SL, all associated partitions are loaded into the MATLAB workspace as MATLAB structures. When you use SB2SL to convert a SuperBlock into Simulink, you must select the partition from which to reference the data for building the Simulink model. To choose the desired data partition, select **Partition** from the **Build** menu on the main SB2SL GUI.

This opens the following window.



Select the partition you want your Simulink model to use, and press **Apply**.

## Converting SuperBlocks to Simulink Models

Before converting your SuperBlock to a Simulink model, there are several options you can set for building the model and recording the translation. See “Setting Translation Options” for more information.

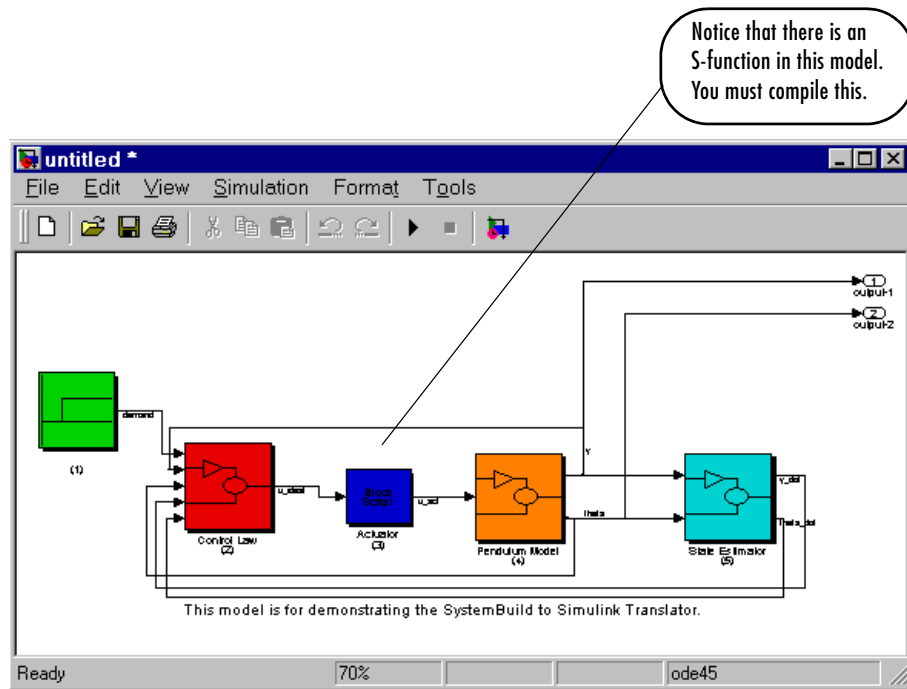
You are ready to convert your model to Simulink once you have:

- Selected the top level SuperBlock and the partition you want to translate
- Set any desired translation options

To begin the translation, select the **Convert** button on the main SB2SL GUI. This begins the translation process and the resulting Simulink model is opened when it is finished. During the translation:

- The progress bar beneath the **Convert** button on the main SB2SL GUI slides toward completion.
- The message window displays actions describing the translation.

Once you convert your model to Simulink, some blocks on the Simulink diagram may be labeled Unconverted. See “Blocks Not Converted to Simulink” and “Suggestions for Handling Unconverted Blocks” for information about Unconverted blocks.

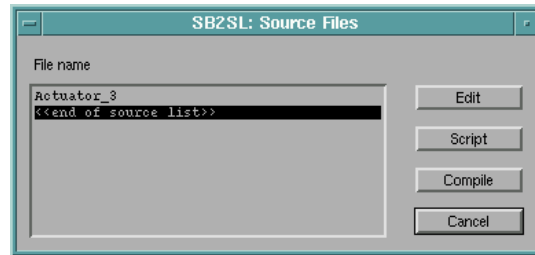


**Figure 3: Simulink Model for sbpend.xmd**

## Compiling Converted BlockScript

When you convert using SB2SL, SystemBuild BlockScript blocks are translated into C code and placed into Simulink S-functions automatically. Choose **Compile** under the **Build** menu of the main SB2SL GUI to open the

**Source Files** window. This window lists the S-functions generated by the translated SystemBuild BlockScript.



From **Source Files** window:

- Select the files you want compiled
- Select the **Compile** button, and MATLAB's standard mex command compiles these C code S-functions.

For more information on MATLAB's MEX-file capability and Simulink S-functions, see the *MATLAB Application Program Interface Guide* or *Writing S-Functions*.

## Saving Translated Models and Data

Once the translation is complete, select the **Save** submenu under the **File** menu in the main SB2SL GUI to save either your model or your data:

- Select **Model** under the **Save** submenu to save the Simulink model to a file so that it can be reloaded directly from MATLAB and Simulink.
- Select **Data** under the **Save** submenu to save the model data read from the SystemBuild file during the translation.

---

**Note** You can set the `PreLoadFcn` callback on the Simulink block diagram to reload the model data file the next time the Simulink model is opened. See *Using Simulink* for details on model callbacks.

---

## Generating a Report

You can generate a report recording the details of your translation after you convert a model with SB2SL. There are several options you may want to set beforehand. See “Report Generation Options” for information on these options.

To generate a report with the default option settings, select **Report** from the **Build** menu after converting your model.

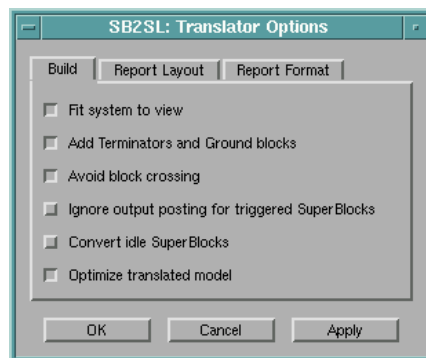
## Setting Translation Options

Before you convert your SystemBuild model to Simulink, you can set options for:

- Building the Simulink models
- Generating reports from the translation
- Converting the reports to various text formats

## Translation Build Options

To set the translation build options select the **Options** menu under **Build** in the main SB2SL GUI.



You have the following build options available:

- **Fit system to view**
  - Select this option to scale the model to fit the window size.
  - Don't select this option if you want to use the original block sizes.

- **Add Terminator and Ground blocks**

Select this option to terminate unconnected block inputs or outputs with Simulink Terminator or Ground blocks.

- **Avoid block crossing**

Select this option to minimize crossing blocks with signal lines in the Simulink model resulting from SB2SL translation.

- **Ignore output posting for triggered SuperBlocks**

When you select this option:

- All triggered SystemBuild outputs are posted in “as soon as finished (SAF)” mode.
- Triggered SuperBlocks assigned to “after timing requirement (ATR)” and “at next trigger (ANT)” output posting modes are ignored.

- **Convert idle SuperBlocks**

- If any enabled and/or triggered SuperBlocks in your model are nested, one or more of the SuperBlocks may never be executed.
- You can choose to convert these idle SuperBlocks by selecting this option.
- The default mode of operation is not to convert idle SuperBlocks.

- **Optimize translated model**

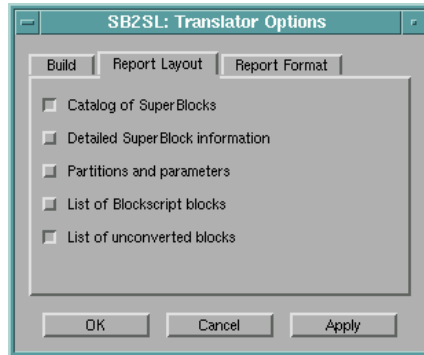
When you select this option, SB2SL maximizes the use of standard Simulink blocks when translating the following SystemBuild blocks:

- Data store blocks
- Algebraic/Logical expression blocks
- Integrator blocks



## Report Generation Options

You can use report generation options to select those portions of the SystemBuild data you want to include in a report. Select the **Report** tab on the **Translator Options** window to access these options.

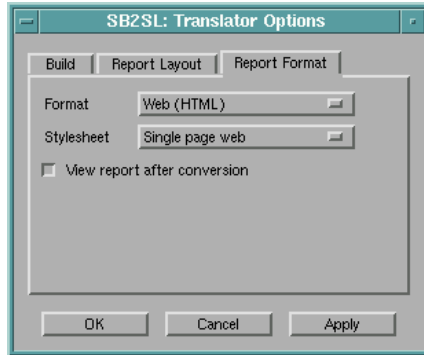


## Report Formatting Options

You have the following options for specifying the format of generated reports:

- Output format (HTML, RTF, or TeX)
- Style
  - Single or multipage output for the HTML format
  - Standard, simple, or large type print for the RTF or TeX formats
- A checkbox you can select to display the report once it is generated

Select the **Report Format** tab on the **Translator Options** window to access these options.



## Compatibility Between SystemBuild and Simulink

SB2SL performs a block-by-block translation of the SystemBuild model. For SystemBuild blocks for which a clear equivalent exists in Simulink, SB2SL places the equivalent built-in Simulink block into the resulting Simulink model. The Gain block is an example in which there is a clear equivalent between SystemBuild and Simulink blocks.

There are other SystemBuild blocks for which there are no clear Simulink equivalents. However, through the use of Simulink's masking and library features, equivalent implementations of these blocks have been created and are found in a special Simulink library called `libsb2s1.mdl`.

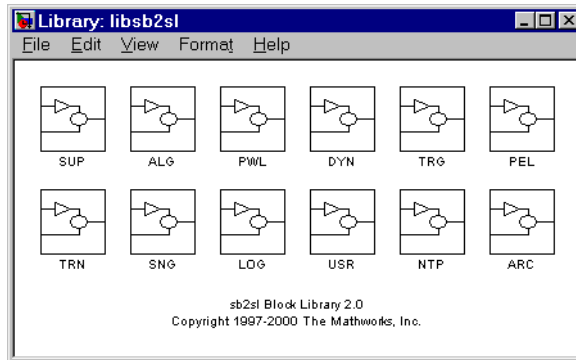
An example of this type of block is the Ramp block found in the SystemBuild SNG library. This block supports limits on the output and a relative start time for the ramp. The standard Simulink Ramp block does not inherently support these features. SB2SL translates this block into a masked subsystem that includes a collection of existing Simulink blocks. This masked subsystem behaves the same as the SystemBuild Ramp block.

### SB2SL Simulink Library

You can find all of the masked blocks generated by SB2SL that are not found in any of the other Simulink libraries in the library `libsb2s1.mdl`. You can open this library at the MATLAB command line by typing

```
libsb2s1
```

After using SB2SL for translation, some of the blocks appearing in the resulting Simulink model may be from this library.



Open any of the masks of the Simulink blocks in this library to see the exact implementation of each of the SystemBuild equivalents used by SB2SL. For example, the Simulink equivalent to the SystemBuild Ramp block is found in

`libsb2sl/SGN/LimRamp`

## Using Real-Time Workshop with Converted SB2SL Models

You can use the Real-Time Workshop® 4.0 to generate code for models you have converted from SystemBuild to Simulink (using SB2SL). Code is generated for most translated blocks in the model.

The blocks that do *not* support code generation through the Real-Time Workshop 4.0 are:

- ConditionBlock
- Decoder
- Encoder
- GainScheduler
- Interp Table (Archive library)
- ShiftRegister

## Limitations

No translator can completely convert an *optimally* designed SystemBuild model into a *optimally* designed Simulink model. There are subtle differences in the way that the two tools work that prevent faithful translation of all capabilities. However, this tool does the job of converting basic blocks and hierarchy from one tool to the other in a form that can be simulated. The following are limitations of SB2SL:

- Data types are not supported.
- Write to Variable and Read from Variable blocks do not support the element- or bit-addressing option.
- The SystemBuild simulation parameter `cdelay` is not supported.
- The timing of triggered subsystems with the “as soon as finished” output posting requirement differs from the SystemBuild implementation:
  - SystemBuild updates the outputs at the beginning of the next minor numerical integration step.
  - In Simulink, the outputs are available immediately.
- Simulink models obtained from SB2SL conversions of SuperBlocks containing any triggered SuperBlocks with both of the following attributes will not run:
  - The output posting is selected as “at timing requirement.”
  - The triggered SuperBlock is nested within another triggered SuperBlock.

### File Formats

SB2SL cannot read SystemBuild files stored in the binary file format.

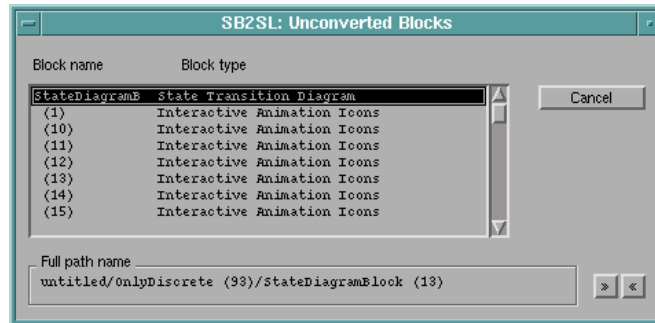
### Blocks Not Converted to Simulink

SB2SL converts the following SystemBuild blocks into empty placeholder blocks in Simulink. You may want to replace these with various Simulink blocks you have developed that are equivalent.

- State Transition Diagrams
- MathScript blocks
- UserCode blocks

- Interactive Animation blocks
- Any new blocks introduced since SystemBuild Version 6.0.

These blocks are converted into blocks labeled Unconverted. To view a complete listing of the blocks not translated, select **Unconverted blocks** from the **Build** menu.



### Suggestions for Handling Unconverted Blocks

You can implement all of the SystemBuild operations represented by the Unconverted blocks on your Simulink diagram using MATLAB, Simulink, and, in some cases, other related products. Here are some suggestions for replacing the Unconverted blocks with ones usable for simulation with Simulink:

- You can replace MathScript blocks with MATLAB Fcn blocks. These blocks are used to run M-files. You must write your own M-files to execute the equivalent MathScript.
- You can replace UserCode blocks with S-Function blocks. These are blocks you can use to run C code or Fortran.
- You can use a variety of blocks in the Simulink Sinks library to replace Interactive Animation blocks, depending on the function of that block. For a greater variety of animated blocks, see the Gauges Blockset.
- You can replace State Transition Diagrams with Stateflow<sup>®</sup> models. This requires you to purchase Stateflow in addition to MATLAB and Simulink.

To replace an Unconverted block in your Simulink model with the correct Simulink block:

- 1** Open an Unconverted block in the Simulink model by double-clicking on it.

This opens a window listing the SystemBuild component that caused the Unconverted block to be created.

- 2** Either:

- a** Delete the Unconverted block and copy an appropriate standard Simulink block into its place.
- b** Use the Simulink function `replace_block` to replace the Unconverted block in the Simulink model.





## B

- BlockScript
  - compiling 9
  - converting 9
- Build menu
  - Compile 9
  - Partition 7
  - Report 11
  - Unconverted blocks 18
- build options 11

## C

- compatibility 15
- compiling BlockScript 9
- converting
  - blocks not converted 18
  - BlockScript 9
  - models to Simulink 8, 17
  - SuperBlocks 17

## F

- file manager 6
- format
  - model requirements 17
  - reports 13
  - SystemBuild models 4

## G

- generating
  - models 8
  - reports 11

## I

- installation 3

## L

- library 15
- limitations 4, 17
- loading a model 5

## M

- main GUI 5
- model tree structure 6
- models
  - format 17
  - generating 8
  - loading 5
  - saving 10

## O

- opening SB2SL 4
- options
  - build 11
  - reports
    - format 13
    - generating 13
  - translation 11

## P

- partitions, selecting 7

**R**

- replace\_block 19
- reports
  - format 13
  - generating 11, 13
  - options 13
  - specifying 13
- requirements 2

**S**

- saving
  - model data 10
  - Simulink models 10
- SB2SL
  - blocks not converted 18
  - blocks, not converted 17
  - compatibility with SystemBuild 15
  - conversion 8
  - installation 3
  - limitations 4
  - main GUI 5
  - models, loading 5
  - opening 4
  - requirements 2
  - Simulink library 15
  - Windows menu, tree 6
- SB2SL translation 8
- selecting
  - partitions 7
  - SuperBlocks 6
- Simulink library 15
- Simulink models
  - options 11
  - saving 10
- Source Files window 10
- specifying reports 13

**SuperBlocks**

- conversion 8
- file manager 6
- partitions, selecting 7
- selecting 6
- translation 8
- tree structure 6

**SystemBuild models**

- format 4, 17

**T**

- translation
  - options 11
  - SuperBlocks, of 8

**W**

- Windows menu, Tree 6