Release Notes for Simulink® Design Verifier™

How to Contact MathWorks

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`www.mathworks.com/contact_TS.html`	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

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

Release Notes for Simulink® Design Verifier™

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Contents

R2011a

R2010bSP1

R2010b

R2010a

R2009bSP1

R2009b

R2009a

R2008b

R2008a+

R2008a

R2007b+

R2007b

R2007a+

R2013a

Version: 2.4
New Features: Yes
Bug Fixes: Yes

Detection of out-of-bound array access design errors

Before you simulate a model, you can use design error detection analysis to find out of bound array access errors. To detect out of bound array access errors in your model, from the Simulink® Editor, select **Analysis > Design Verifier > Options**. In the Configuration Parameters dialog box, on the **Design Verifier > Design Error Detection** pane, select **Out of bound array access**. On the **Design Verifier** pane, click **Detect Errors**.

For more information, see “Detect Out of Bound Array Access Errors”.

R2012b

Version: 2.3
New Features: No
Bug Fixes: Yes

R2012a

Version: 2.2
New Features: Yes
Bug Fixes: Yes

Design Error Detection For Dead Logic

You can now use design error detection to find dead logic in your model. Simulink Design Verifier™ uses multiple analysis engines, including Polyspace® and Prover®, to identify the dead logic. Previously, to uncover dead logic in your model, you had to analyze the results of test case generation. To conduct design error detection analysis, in the Configuration Parameters dialog box, on the **Design Verifier > Design Error Detection** pane, select **Dead Logic**. See Detecting Dead Logic.

Filtering Model Objects From Model Coverage

Simulink Design Verifier now allows you to filter certain model objects from model coverage during test case generation. The objects that you specify to exclude are stored in an external file. In the Configuration Parameters dialog box, on the **Design Verifier > Test Generation** pane, select **Ignore objectives based on filter**. In the **Coverage filter file** field, enter the file name.

For more information about coverage filtering, see Excluding Model Objects From Coverage in the Simulink Verification and Validation™ documentation.

Improved Property Proving For Look-Up Tables

Simulink Design Verifier now automatically optimizes property proving for Look-Up Table blocks, improving performance.

R2011b+

Version: 2.1.1
New Features: Yes
Bug Fixes: Yes

sldvtimer Function Available For Generating Test Cases

In R2011b+, the `sldvtimer` function is available to identify, change, and display timer optimizations. When generating test cases, you can use the function to:

- Identify where you can apply timer optimization in your model.
- Determine if Simulink Design Verifier applied timer optimizations during test generation.
- Configure timer optimizations.

R2011b

Version: 2.1
New Features: Yes
Bug Fixes: Yes

Checking Specified Design Minimum and Maximum Values

If you have specified minimum and maximum values on your Simulink model, the Simulink Design Verifier software can analyze your model to make sure that the intermediate and output signals do not violate the specified values. If the analysis finds a violation, the software creates a test case that demonstrates the violation.

This check is a new option for the design error detection analysis mode. For detailed information, see Checking for Specified Intermediate Minimum and Maximum Signal Values.

Improved Support for Trigonometric Functions

For a test case generation analysis, the `CombinedObjectives (Nonlinear)` and `LargeModel (Nonlinear Extended)` test suite optimizations have improved support for trigonometric and nonlinear math functions.

Improved Support for Large Lookup Tables

All the Simulink Design Verifier analysis modes have improved support for large lookup tables.

Optimized Handling for Extending Existing Test Cases

Existing test cases are now represented in a more memory-efficient way internally in the test generation engine. This improvement makes it possible to extend significantly longer existing test cases with more time steps.

Support for Trigger and Enable Ports for Model Blocks

The Simulink Design Verifier software fully supports root-level Trigger and Enable ports for referenced models.

Changed Format for sldvruntest and sldvruncgvtest Output

Compatibility Considerations: Yes

The output format for `sldvruntest` and `sldvruncgvtest` has changed in R2011b. In R2011b, the output argument contains the following data for each test case executed in an array of `Simulink.SimulationOutput` objects.

Field	Description
`tout_sldvruncgvtest`	Simulation time
`xout_sldvruncgvtest`	State data
`yout_sldvruncgvtest`	Output signal data
`logsout_sldvruncgvtest`	Signal logging data for: • Signals that are connected to outports • Signals that are configured for logging on the model

Compatibility Considerations

If you have scripts that depend on the output from `sldvruntest` and `sldvruncgvtest`, you can temporarily specify the output format. Use the nonvisible field `outputFormat` in the `runOpts` structure that `sldvruntestopts` creates as follows:

```
runOpts = sldvruntestopts;
runOpts.outputFormat = 'TimeSeries';
sldvruntest(model_name, sldvData, runOpts);
```

or

```
runOpts = sldvruntestopts;
runOpts.outputFormat = 'StructureWithTime';
sldvruntest(model_name, sldvData, runOpts);
```

Conversion of Error and Warning Message Identifiers

Compatibility Considerations: Yes

For R2011b, error and warning message identifiers have changed in Simulink Design Verifier.

Compatibility Considerations

If you have scripts or functions that use message identifiers that changed, you must update the code to use the new identifiers. Typically, message identifiers are used to turn off specific warning messages, or in code that uses a `try/catch` statement and performs an action based on a specific error identifier.

For example, the `SLDV:InvalidConfSet` identifier has changed to `SLDV:configcomp_get:InvalidConfSetRef`. If your code checks for `SLDV:InvalidConfSet`, you must update it to check for `SLDV:configcomp_get:InvalidConfSetRef` instead.

To determine the identifier for a warning, run the following command just after you see the warning in the MATLAB® command window:

```
[MSG,MSGID] = lastwarn;
```

This command saves the message identifier to the variable `MSGID`.

To determine the identifier for an error that appears at the MATLAB prompt, run the following commands just after you see the error:

```
exception = MException.last;
MSGID = exception.identifier;
```

Note Warning messages indicate a potential issue with your code. While you can turn off a warning, a suggested alternative is to change your code so it runs warning-free.

R2011a

Version: 2.0
New Features: Yes
Bug Fixes: Yes

Automatic Detection of Overflow and Divide-by-Zero Design Errors

If your Simulink model performs arithmetic operations, the Simulink Design Verifier software can analyze the model to identify design errors that occur at run time. The analysis detects two types of errors:

- Integer or fixed-point data overflow
- Division by zero

After the analysis is complete, you can create a harness model that includes test cases for each error.

For more information, see Detecting Integer Overflow and Division-by-Zero Errors.

Improved Analysis Results Workflow

After you analyze a Simulink model using the Simulink Design Verifier software, you can choose how you want to review the results. You have the following options:

- Highlight the analysis results on the model.
- Generate the detailed analysis report.
- Create the harness model with the generated test cases in the Signal Builder.
- Simulate the test cases and produce a model coverage report (test case generation only).

Improved Support for Nonlinear Arithmetic and Math Operations

The Simulink Design Verifier test-generation software includes improved support for nonlinear arithmetic and math operations that:

- Improves test generation for models containing nonlinear operations.
- Improves support of math operations such as trigonometric functions.
- Improves scalability to very large models.

To use these new strategies, in the Configuration Parameters dialog box, on the **Design Verifier > Test Generation** pane, set the **Test suite optimization** parameter to one of the following:

- `CombinedObjectives (Nonlinear Extended)`
- `LargeModel (Nonlinear Extended)`

New Capability to Highlight Analysis Results on the Model

When a Simulink Design Verifier analysis is complete, you can specify that the software use color highlighting on the model to indicate the analysis results for individual objects. When you click an object, you see the detailed results specific to that object.

You can generate a detailed analysis report or create a harness model that contains test case signals at any time.

After you run a design error detection analysis, the model is highlighted by default. After you run a test case generation or property-proving analysis, you can highlight the model.

For more information, see Highlighted Results on the Model.

New Capability to Review Model Analysis Results in Model Explorer

If you close the Simulink Design Verifier analysis results so you can fix the causes of problems in your model, you might need to review the analysis results again. As long as your model remains open, you can view the results of your most recent analysis results in the Model Explorer by selecting **Tools > Design Verifier > Latest Results**.

From the Model Explorer, you can:

- Highlight the analysis results on the model.
- Generate the detailed analysis report.
- Create the harness model with the generated test cases in the Signal Builder.
- Simulate the test cases and produce a model coverage report (test case generation only).

For more information, see Reviewing Analysis Results in the Model Explorer.

New Temporal Operator Blocks

The Simulink Design Verifier block library includes three new blocks that allow you to define temporal properties on Boolean signals in your model:

- Detector — Detect true duration on input and construct output true duration based on output type
- Extender — Extend true duration of input
- Within Implies — Capture within implication if observed input is `true` within each true duration of first input

Support for Simulink Blocks

The Simulink Design Verifier software now supports the following Simulink blocks:

- Discrete Derivative
- Function-Call Feedback Latch (new block)
- Probe
- Rate Limiter Dynamic
- Trigonometric Function — Supported when **Function** is `sin`, `cos`, or `sincos` and **Approximation method** is `CORDIC`.
- Variant Subsystem
- Weighted Sample Time
- Weighted Sample Time Math

R2010bSP1

Version: 1.7.1
New Features: No
Bug Fixes: Yes

R2010b

Version: 1.7
New Features: Yes
Bug Fixes: Yes

Support for 64-Bit Windows Operating Systems

The Simulink Design Verifier software can now execute in 64-bit mode on 64-bit Windows® systems.

New Support for Specified Input Minimum and Maximum Values as Analysis Constraints

The Simulink Design Verifier software can consider the specified minimum and maximum values for input signals during a test case generation or property proving analysis. This feature allows you to:

- Constrain the test cases based on the specified minimum and maximum values.
- Assume the specified minimum and maximum values during property-proving analysis.

New Built-In Support for Automating Test Execution in SIL/PIL Mode via the CGV API

The new `sldvruncgvtest` function allows you to execute test cases in Software-in-the-Loop (SIL) or Processor-in-the-Loop (PIL) mode using the Code Generation Verification (CGV) API methods.

New Support for Extracting and Analyzing Stateflow Atomic Subcharts

The Simulink Design Verifier software now provides support for Stateflow® atomic subcharts. Atomic subcharts make it easier to isolate parts of a Stateflow chart for development and analysis.

The Simulink Design Verifier support for atomic subcharts allows you to:

- Generate test cases and prove properties for an atomic subchart in a Stateflow chart within a Simulink model.
- Use `sldvextract` to extract the contents of an atomic subchart and create a model for the Simulink Design Verifier software to analyze.

New Capability to Eliminate Unused Signals from the Generated Harness

To improve the performance of harness model creation during a Simulink Design Verifier analysis, from the Signal Builder block, the software can omit signals that have no effect on the output of the model.

`sldvmakeharness` supports this capability if you set the `usedSignalsOnly` harness model option to `true`.

Support for Simulink Blocks

The Simulink Design Verifier software now supports the following Simulink blocks:

- Dead Zone and Dead Zone Dynamic
- Lookup Table Dynamic
- Probe (Partial support)
- Width

sldvlogsignals Replaces sldvlogdata

Compatibility Considerations: Yes

The `sldvlogsignals` function replaces the `sldvlogdata` function. Use `sldvlogsignals` to:

- Simulate a Simulink model and in that model, log all the inputs to a specified Model block.
- Simulate all or some of the test cases in a harness model created by:
  - Simulink Design Verifier analysis
  - `sldvmakeharness`
  - `slvnvmakeharness`

Compatibility Considerations

The `sldvlogsignals` function replaces the `sldvlogdata` function. Currently, if you use the `sldvlogdata` function, it automatically redirects to `sldvlogsignals`. Update your scripts to use `sldvlogsignals`.

sldvmergeharness Replaces sldvharnessmerge

Compatibility Considerations: Yes

The `sldvmergeharness` function replaces the `sldvharnessmerge` function. `sldvmergeharness` combines the test cases and initializations from any specified harness models into a single harness model.

Compatibility Considerations

Currently, if you use the `sldvharnessmerge` function, it automatically redirects to `sldvmergeharness`. Update your scripts to use `sldvmergeharness`.

R2010a

Version: 1.6
New Features: Yes
Bug Fixes: Yes

Generate Test Cases for Missing Coverage Data

The Simulink Design Verifier software now offers the option to isolate test generation to objectives that are not satisfied in simulation coverage results. If you simulate your model, but do not achieve 100% coverage, you can analyze the model using the Simulink Design Verifier test-generation capability to find test cases that achieve the missing coverage.

If you select the **Ignore objectives satisfied in existing coverage data** parameter in the Configuration Parameters dialog box, you can import the coverage data file; the analysis eliminates all objectives satisfied in the coverage results.

sldvlogdata Function for Logging Test Cases During Simulation

With the new `sldvlogdata`function, you can:

- Simulate a Simulink model and in that model, log all the inputs to a Model block.
- Simulate all or some of the test cases in a harness model created by the Simulink Design Verifier software and log all the inputs to the test unit.

You can save logged data to a MAT-file and use that file as input to the Simulink Design Verifier software for extending tests. This allows you to generate more realistic test cases and extends the analysis to complete the test suite.

Extend Existing Test Cases

The Simulink Design Verifier software now offers the option to extend existing test cases with additional time steps to generate complete test suites. This allows the software to generate test cases for parts of your model that are hard to analyze.

If you enable the **Extend existing test cases** parameter in the Configuration Parameters dialog box, the software imports the logged test cases from a MAT-file. If you also enable the **Ignore objectives satisfied by existing test cases** parameter, the analysis generates results, ignoring the coverage objectives satisfied by the logged test cases. Otherwise, the analysis efficiently creates a complete test suite.

Demo Library and Models to Support Temporal Properties Specification

The Simulink Design Verifier software includes a new **Temporal Property Specification** demo category that includes:

- A Temporal Operator Blocks demo library that contains the following blocks and examples:
  - Detector — Detects a user-specified length of true duration on the input signal and constructs an output true duration of length based on the output type.
  - Extender — Extends the true duration of the input signal by a fixed number of time steps or indefinitely.
  - Within Implies — Captures the within implication by observing whether the second input is true for at least one time step within each true duration of the first input.
  - Temporal Property Specification examples — A library model that includes examples that use the Detector, Extender, and Within Implies blocks
- Two demo models that contain these blocks:
  - Debounce Temporal Properties
  - Power Window Controller Temporal Properties

Support for Stateflow Absolute-Time Temporal Logic Operators

The Simulink Design Verifier software now supports the Stateflow absolute-time temporal logic operators. For more information, see Operators for Absolute-Time Temporal Logic in the Stateflow documentation.

Support for Simulink Blocks

The Simulink Design Verifier software now fully supports the following blocks:

- Backlash
- Cosine
- Discrete Derivative
- Sine

The Simulink Design Verifier software now provides improved support for the following blocks:

- Interpolation Using Prelookup
- Lookup Table (n-D)

For more information, see Simulink Block Support.

R2009bSP1

Version: 1.5.1
New Features: No
Bug Fixes: Yes

R2009b

Version: 1.5
New Features: Yes
Bug Fixes: Yes

New Functions for Verification Objectives and Constraints

Compatibility Considerations: Yes

Use these four new functions to specify objectives and constraints within an Embedded MATLAB® script. You can use these functions instead of the corresponding Simulink Design Verifier blocks.

Function	Purpose	Corresponding Block
`sldv.assume`	Proof assumption	Proof Assumption
`sldv.condition`	Test condition	Test Condition
`sldv.prove`	Proof objective	Proof Objective
`sldv.test`	Test objective	Test Objective

These functions:

- Identify mathematical relationships for objectives and constraints in a form that can be more natural than using block parameters
- Support specifying multiple constraints without complicating the model
- Provide access to the power of the MATLAB software
- Support separation of verification and model design

Compatibility Considerations

The following functions will be removed in a future release:

- `dv.assume`
- `dv.condition`
- `dv.prove`
- `dv.test`

So that your models with those functions will work in future releases, replace these functions with the corresponding new function added in this release. For example, replace `dv.assume` with `sldv.assume`.

Support for Enumerated Signals and Parameters

The Simulink Design Verifier software now supports Simulink models with enumerations. All the Simulink Design Verifier library blocks support enumerated parameters, constants, and inputs.

New Option to Stop Simulation on Proof Violation

The Simulink Design Verifier software allows you to stop a model simulation if it encounters a property violation. You enable this capability by inserting a Proof Objective block into a model and setting the **Stop simulation when the property is violated** parameter. If the simulation detects a violation of the property specified in the Proof Objective block, it terminates with an error.

Therefore, you can now verify a counterexample that was detected during a Simulink Design Verifier analysis.

New sldvmakeharness Function

With the new `sldvmakeharness` function, you can:

- Create a test harness model from existing Simulink Design Verifier analysis data.
- Create an empty test harness model directly from a Simulink model.

New sldvreport Function

You can now generate and customize a report from existing Simulink Design Verifier analysis data with the new `sldvreport` function.

New Support for Simulink Blocks

The Simulink Design Verifier software now supports the following blocks and parameters:

- Direct Lookup Table (n-D)
- Discrete Transfer Fcn
- Lookup Table and Lookup Table (2-D) — Except when the **Lookup method** block parameter specifies `Interpolation-Extrapolation` and the block's input and output signals do not have the same floating-point data type.
- Math Function — All signal types now support the `hermitian` and `transpose` function parameter settings
- Rate Limiter — For signals of all data types
- Shift Arithmetic — For all parameters and signals of all data types
- Tapped Delay
- Transfer Fcn Direct Form II
- Transfer Fcn Direct Form II Time Varying

Support for New Blocks

The Simulink Design Verifier software supports the following new Simulink blocks:

- Discrete PID Controller
- Discrete PID Controller (2 DOF)
- Enumerated Constant

R2009a

Version: 1.4
New Features: Yes
Bug Fixes: Yes

Automatic Stubbing for Unsupported Operations

Automatic stubbing allows you to complete a test-generation or property-proving analysis even if the model contains blocks or functions that the Simulink Design Verifier software does not support, like S-functions and C math operations.

By default, this feature is unavailable. To enable automatic stubbing before running an analysis, on the Configuration Parameters **Design Verifier** main pane, select **Automatic stubbing of unsupported blocks and functions**. In addition, if the compatibility check finds unsupported blocks that automatic stubbing can handle, you can enable automatic stubbing at that time.

Long Test Case Optimization

`Long test cases` is a new option for the **Test suite optimization** parameter. The `Long test cases` option instructs the Simulink Design Verifier software to create fewer but longer test cases that each satisfy multiple test objectives. With this option, you can customize the analysis results, run a more efficient analysis, and create easier-to-review results, in both Signal Builder and in the HTML report that the software generates.

New Support for Blocks

The Simulink Design Verifier software now supports models containing the following blocks:

- Combinatorial Logic
- Decrement Time To Zero
- Discrete Filter
- Fixed-Point State-Space
- Integer Delay
- Model blocks that reference other models
- Prelookup
- Relay

Analyzing External Functions for Embedded MATLAB Function Blocks

If your model contains an Embedded MATLAB Function block that calls any external functions, the Simulink Design Verifier software can now accumulate coverage results for those functions.

Enhanced Block Replacement Capability for Subsystems and Model Blocks

You can write your own replacement rules to replace subsystem or Model blocks that reference another model with the Simulink Design Verifier block replacement capability. The software replaces a subsystem or Model block with a different subsystem or with a built-in block as defined in the block replacement rules.

New Implies Block

The new Implies block simplifies property specification. You can now specify conditions that produce a given response. For example, you can quickly create expressions indicating that pressing the brake pedal implies the cruise control must be inactive.

You can use the Implies block in any model, not just when running the Simulink Design Verifier software.

New Property-Proving Examples and Demos

The Simulink Design Verifier block library includes four new example models that demonstrate how to define complex properties for property-proving analysis.

In addition, the following demo models are shipping with R2009a:

- `sldvdemo_sbr_design.mdl` — Finding property violations
- `sldvdemo_sbr_verification.mdl` — Proving that properties are valid
- `sldvdemo_thrustrvs_verification.mdl` — Analyzing model and properties to prove correctness or to identify counterexamples
- `sldvdemo_cruise_control_fxp_verification.mdl` — Proving properties for fixed-point arithmetic with block replacements
- `sldvdemo_cruise_control_verification.mdl` — Supporting model reference and verification subsystems

sldvisactive Function

The `sldvisactive` function checks whether the Simulink Design Verifier software is actively translating the model. This function is called from the masked initialization of masked subsystems and other model or block callbacks to configure the model for Simulink Design Verifier analysis.

For example, the mask initialization of the Environment Controller block invokes the `sldvisactive` function to output the signal at its Sim port when you start analyzing a model that contains the block.

R2008b

Version: 1.3
New Features: Yes
Bug Fixes: Yes

Simulink Bus Signals and Bus Objects Support

Simulink Design Verifier now supports Simulink buses and bus objects:

- The root Inport and Outport blocks accept bus signals.
- Nonvirtual buses are propagated through the model elements.
- The test harness model reconstructs the bus signals from the underlying bus elements.

Fixed-Point Data Support

Simulink Design Verifier blocks now support fixed-point parameters and inputs. These blocks include:

- Test Condition
- Test Objective
- Assumption
- Proof Objective

The `Slvd.Point` and `Sldv.Interval` constructors now accept fixed-point data.

Generating Test Harness Model with Model Reference

To use this option, select **Reference input model in generated harness** in the **Design Verifier > Results** pane of the Configuration Parameters dialog box. Simulink Design Verifier software then uses model reference to run the original model from the test harness.

Generating SystemTest TEST-File

To use this option, select **Save test harness as SystemTest TEST-File** in the **Design Verifier > Results** pane of the Configuration Parameters dialog box. The software creates a TEST-file instead of a test harness model. Using a TEST-file allows you to run the test cases in the SystemTest™ environment and configure the model coverage settings using the SystemTest software.

Improved Search Algorithms

This release includes search algorithms for the following two modes that improve the performance and the quality of the results:

- Test case generation — The combined objectives options minimizes the number of test cases by generating cases that address more than one test objective.
- Property proving — Proving that model properties are valid.

New Data File Format

Compatibility Considerations: Yes

When the Simulink Design Verifier software completes an analysis, it creates a data file. Now the data file supports bus input ports and includes more information about the analyzed model. For more information, see Simulink Design Verifier Data Files in the Simulink Design Verifier documentation.

Compatibility Considerations

To convert an `sldvData` structure from the old format to the new format, use the `Sldv.DataUtils.convertToCurrentFormat` utility with the following syntax:

```
new_sldvData = Sldv.DataUtils.convertToCurrentFormat(model, old_sldvData)
```

The arguments used for this conversion comprise:

- `model` — The name of the model that was analyzed
- `old_sldvData` — The name of an `sldvData` structure created using the old (pre-R2008b) format

To convert an `sldvData` structure in the new format to the old format, use the `Sldv.DataUtils.convertToOldFormat` utility with the following syntax:

```
old_sldvData = Sldv.DataUtils.convertToOldFormat(model, new_sldvData)
```

The arguments used for this conversion comprise:

- `model` — The name of the model that was analyzed
- `new_sldvData` — The name of an `sldvData` structure created using the format that is new with R2008b

New HTML Report

The HTML report that Simulink Design Verifier software generates has been enhanced. Now, when you select **Generate report of the results** in the **Design Verifier > Report** pane of the Configuration Parameters dialog box, the report generated has several improvements:

- The report generates faster and is easier to understand.
- The report can display expected outputs.
- The software generates a report that reflect the analysis settings (for example, test case generation vs. property proving).

Blocks with No Input Ports Limitation

If a Simulink model has any blocks with no input ports, Simulink Design Verifier software cannot generate the test harness.

R2008a+

Version: 1.2.1
New Features: No
Bug Fixes: Yes

R2008a

Version: 1.2
New Features: Yes
Bug Fixes: Yes

Embedded MATLAB Subset Support

This release provides support for the Embedded MATLAB Function block in the Simulink software and Embedded MATLAB functions in the Stateflow software. For more information, see Support Limitations for MATLAB for Code Generation in the Simulink Design Verifier User's Guide.

Enhanced Support for Stateflow Truth Tables

Previous releases support only the Stateflow Classic truth tables. However, this release introduces support for Embedded MATLAB truth tables in the Stateflow software, which includes support for the Truth Table block. See Truth Table Functions for Decision-Making Logic in the Stateflow documentation for more information.

New Simulink Design Verifier Data File Options

This release introduces new options on the **Design Verifier > Results** pane of the Configuration Parameters dialog box:

- **Include expected output values** — Simulates the model using the test case signals and includes the output values in the Simulink Design Verifier data file.
- **Randomize data that does not affect outcome** — Assigns random values instead of zeros to input signals that have no impact on test or proof objectives.

New Test Suite Optimization Setting

In this release, the **Test suite optimization** parameter that appears on the **Design Verifier > Test Generation** pane of the Configuration Parameters dialog box includes a new setting: `Large model`. This test generation strategy is tailored to large, complex models that contain nonlinearities and many test objectives.

R2007b+

Version: 1.1.1
New Features: No
Bug Fixes: Yes

R2007b

Version: 1.1
New Features: Yes
Bug Fixes: Yes

Fixed-Point Data Type Support

This release provides support for fixed-point data types. For more information, see Fixed-Point Support Limitations in the Simulink Design Verifier User's Guide.

R2007a+

Version: 1.0
New Features: Yes
Bug Fixes: No

Introducing the Simulink Design Verifier Software

The Simulink Design Verifier software extends the Simulink and Stateflow products with formal methods that help you confirm your model and chart behavior. The Simulink Design Verifier software performs a mathematically rigorous analysis of your model to identify all of its possible execution pathways. Subsequently, the software can

- **Generate Tests**

  The Simulink Design Verifier software can generate tests that satisfy your model's coverage objectives, including decision coverage, condition coverage, and modified condition/decision coverage (MC/DC). You can even customize the tests that it generates by using Simulink Design Verifier blocks that allow you to specify your own objectives and to constrain signal values. After the software completes its analysis, it produces a test harness model with a Signal Builder block that contains test signals. Simply simulate the test harness model to confirm that the test signals achieve your model's objectives.

- **Prove Properties**

  The Simulink Design Verifier software can prove that signals in your model attain particular values or ranges. Use Simulink Design Verifier blocks to specify values and ranges that you desire signals to attain, or to constrain the values of other signals. If the software disproves any of the values or ranges given the constraints you specify, it produces a test harness model with a Signal Builder block that contains signals comprising counterexamples. Simply simulate the test harness model to confirm that the counterexamples falsify your model's properties.

The Simulink Design Verifier software documents its analysis results in an HTML report. Also, it produces a data file containing the analysis results, which you can postprocess for your own analyses and reports.

In short, the Simulink Design Verifier software gives you confidence in the behavior of your Simulink models and Stateflow charts.

Version 1.0 of the Simulink Design Verifier software was released in a Web-downloadable form after R2007a.