Simulink® Verification and Validation™ Release Notes

MATLAB® & SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news:	`www.mathworks.com`
Sales and services:	`www.mathworks.com/sales_and_services`
User community:	`www.mathworks.com/matlabcentral`
Technical support:	`www.mathworks.com/support/contact_us`
Phone:	508-647-7000

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

Simulink® Verification and Validation™ Release Notes

Trademarks

Patents

Contents

R2017a

R2016b

R2016a

R2015aSP1

R2015b

R2015a

R2014b

R2014a

R2013b

R2013a

R2012b

R2012a

R2011b

R2011a

R2010bSP1

R2010b

R2010a

R2009bSP1

R2009b

R2009a

R2008b

R2008a+

R2008a

R2007b+

R2007b

R2007a+

R2007a

R2006b

R2006a

No New Features or Changes

R2017a

Version: 3.13

New Features

Bug Fixes

Compatibility Considerations

Masking Modified Condition Decision Coverage (MCDC) Analysis: Use masking MCDC to achieve increased model coverage

Simulink® Verification and Validation™ now uses the masking modified condition and decision coverage (MCDC) definition for MCDC analysis by default. Masking MCDC accounts for the masking of conditions in subexpressions, allowing for an increased number of MCDC objectives that are satisfied compared to the unique-cause definition of MCDC that Simulink Verification and Validation previously used by default. For more information on the differences between masking MCDC and unique-cause MCDC, see "Modified Condition and Decision Coverage (MCDC) Definitions in Simulink Verification and Validation".

Compatibility Considerations

When you use the masking MCDC definition, some Simulink models that previously received less than complete MCDC coverage now receive increased coverage. For example, cases where some inputs to a complex logic expression cannot change independently are now able to achieve full coverage when you use the masking MCDC definition but not when you use the unique-cause MCDC definition. For more information, and for instructions on how to use the unique-cause definition in MCDC analysis, see "Setting the (MCDC) Definition Used for Simulink Verification and Validation Coverage Analysis".

Automatic Dead Logic Justification: Exclude unachievable coverage detected by Simulink Design Verifier

You can now add filtering rules to automatically justify dead logic found during Simulink Design Verifier™ dead logic analysis. The coverage report lists the justified elements in the Objects Filtered from Coverage Analysis section. Justified outcomes are highlighted in light blue in the Details section of the Coverage Report.

You can detect dead logic by using Simulink Design Verifier and add justification rules for all dead logic from the Filter tab in the **Current Cumulative Data** section in the **Coverage Results** dialog box. You can also add justification rules for individual objectives from the Details sections of the Coverage Report.

Clone Detection Improvements: Refactor repeating library patterns and subsystem clones

In R2017a, use the Identify Modeling Clones tool to identify more types of clones. Create models that replace clones with library subsystems. Replacing clones with links to library blocks enables component reuse.

- Identify clones across a model hierarchy that includes inactive variants and commented-out model regions.
- Identify modeling patterns that are clones of library subsystems.
- Identify clones that have different parameter settings. You can create a masked library subsystem from these clones and replace each clone instance with a link to the masked library block.

If you have Simulink Coder™ or Embedded Coder® software, you can generate reusable code for library subsystems. See "Enable Component Reuse with Clone Detection".

Stateflow Edit-Time Checks for MAAB: Fix guideline violations as you edit

In R2017a, these MAAB checks have been extended to edit-time checking for Stateflow charts:

- **Check entry formatting in State blocks in Stateflow charts**: Identify missing line breaks between entry action (`en`), during action (`du`), and exit action (`ex`) entries in states. Identify missing line breaks after semicolons (`;`) in statements.
- **Check transition actions in Stateflow charts**: Identify missing line breaks between transition actions.
- **Check transition orientations in flow charts**:
  - Draw transition conditions horizontally.
  - Draw transitions with a condition action vertically.
- **Check for MATLAB expressions in Stateflow charts**: Identify Stateflow objects that use MATLAB® expressions that are not suitable for code generation.

Model Metrics API: Analyze referenced models and libraries and aggregate metric data

Metric Algorithms Include Referenced Models and Libraries

When collecting metrics for a top model, metric algorithms now include referenced models and libraries in the analysis. You can also aggregate metrics through these referenced models and libraries. The slmetric.Engine class now includes the following properties:

- `AnalyzeLibraries` (defaults to `true`)
- `AnalyzeModelReferences` (defaults to `true`)

There are the following limitations:

- Custom authored metric algorithms do not follow library links.
- Metric algorithms do not analyze external MATLAB files.

Aggregation by Components

`slmetric.Engine.getMetrics` returns results aggregated by components. Components for model metrics are:

- `Chart`
- `MATLABFunction`
- `MFile`
- `Model`
- `ModelBlock`
- `ProtectedModel`
- `SubSystem`

New slmetric.Engine Functions

The model metric API includes new functions:

- `slmetric.Engine.getMetricDistribution`: for a given metric, generate distribution data

- slmetric.Engine.getStatistics: for a given metric, generate maximum, minimum, mean, and standard deviation of the metric value
- slmetric.Engine.getAnalysisRootMetric: for a given metric, query metric results from a given root

Change to Lines of code for Stateflow blocks metric

In R2017a, metric, “Lines of code for Stateflow blocks metric”, now counts only effective lines of code. Both C and MATLAB action language are now supported.

New Count Metric: Inport outport metric

R2017a introduces a new count metric, “Input output metric”, that measures the number of inports and outports in a component.

Compatibility Considerations

- When generating new metric database files, the metric engine renames and saves metric database files created prior to R2017a.
- Previously, if a metric returned detailed results (or multiple results) per component, getMetrics returned detailed results along with the aggregated results. In R2017a, by default, getMetrics returns only metric data on a component level. If a metric returns detailed results, to access those results, specify AggregationDepth set to None when you call getMetrics.

IBM Rational DOORS Next Generation Support: Link and trace model elements to requirements in DOORS Next Generation

You can now link model elements and establish traceability with requirements in IBM® Rational® DOORS® Next Generation. For more information, see “Requirements Traceability with IBM Rational DOORS Next Generation”.

Model Advisor Check for Nondefault Block Attributes: Customize check and automatically fix warnings

Model Advisor MAAB check **Check for Nondefault Block Attributes** (**Check ID**: mathworks.maab.db_0140) identifies blocks that use nondefault block parameters. In R2017a, you can customize the check and fix warnings.

Task	Action	See
Customize list of nondefault block parameters that the check flags.	In the Model Advisor Configuration Editor, use the **Input Parameters**.	“Customize Model Advisor Check for Nondefault Block Attributes”
Automatically fix check warnings.	After you run the check, in the Model Advisor, select **Add nondefault values into block annotation**.	“Automatically Fix Display of Nondefault Block Parameters”

Library and mask scoping control for MAAB checks

You can now control whether Model Advisor checks in **By Task** > **Modeling Standards for MAAB** > **Simulink** look under masks or follow links. To customize a check to look under masks or follow links for an open model:

1. Navigate to **Model Advisor** > **Settings** > **Open Configuration Editor**.
2. Select **By Task** > **Model Standards for MAAB** > **Simulink** > **Check for Simulink diagrams using nonstandard display attributes**.
3. On the right pane, under **Input Parameters**, you can specific to either **Follow links** or **Look under masks**
4. For the **Follow links** or **Look under masks** settings to take effect, you must save the configuration file (**File** > **Save As**), close the Configuration Editor, and load the saved configuration file in the Model Advisor from **Settings** > **Load Configuration**.

See `find_system` for more information on options for each parameter.

Model Advisor MISRA C:2012 guideline updates

The following guidelines have been combined into this guideline: **Length of user-defined object names to improve MISRA C:2012 compliance**.

- hisl_0063: Length of user-defined function names to improve MISRA C:2012 compliance
- hisl_0064: Length of user-defined type object names to improve MISRA C:2012 compliance
- hisl_0065: Length of signal and parameter names to improve MISRA C:2012 compliance

The following guidelines have been combined into this guideline: **Configuration parameters that improve MISRA C:2012 compliance**.

- hisl_0060: Configuration parameters that improve MISRA C:2012 compliance
- hisl_0312: Specify target specific configuration parameters to improve MISRA C:2012 compliance
- hisl_0313: Selection of bitfield data types to improve MISRA C:2012 compliance

The MISRA check **Check configuration parameters for MISRA C:2012**:

- Now checks configuration parameter `GenerateAllocFcn` to be off. Directive 4.12 requires dynamic memory allocation to not be used.
- No longer checks `MaxIdLength` for a value of 31 (not required by MISRA).

New MISRA check **Blocks not recommended for C/C++ production code deployment** is now separate from the check **Check for blocks not recommended for MISRA C:2012**.

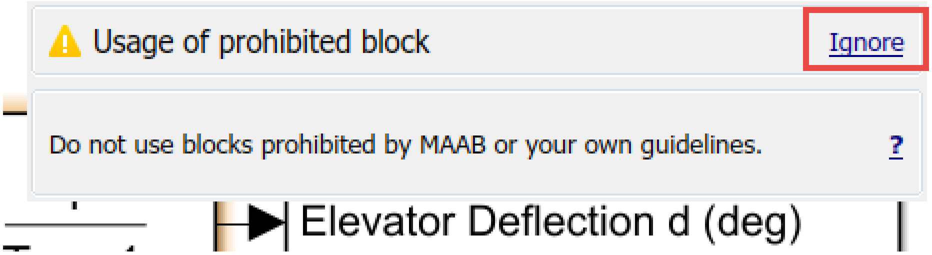
Compatibility Considerations

Blocks not recommended for C/C++ production code deployment has been separated from **Check for blocks not recommended for MISRA C:2012**. Workflows using **Check for blocks not recommended for MISRA C:2012** must include the new check.

Edit-Time Checking: Support for exclusions from Model Advisor

In R2017a, you can exclude checks from edit-time checking.

- Edit-time checking now supports Model Advisor exclusions.
- Exclusions can be saved outside or inside the model.
- There is no difference between adding exclusions for edit-time or for the Model Advisor.
- The **Ignore** button on the warning popup allows users to ignore checks. Clicking the **Ignore** button for the first time launches the exclusion editor.



Do not use blocks prohibited by MAAB or your own guidelines.

For more information, see “Edit-Time Exclusions”.

R2016b

Version: 3.12

New Features

Bug Fixes

Compatibility Considerations

Edit-Time Checking: Detect and fix potential standards compliance issues in your model at design time

Identify standards compliance issues while you edit by using edit-time checking. Edit-time checking provides visual cues for Model Advisor check violations. In the model editor, highlighted blocks alert you to issues with your model. To display information about the issue, hover your cursor over a highlighted block and click the error or warning icon.

To enable edit-time checking for Model Advisor checks, in the model window, select **Analysis** > **Model Advisor** > **Display Advisor Checks in Editor**.

Once you select this option, Model Advisor detects violations for these checks:

- Check subsystem names
- Check port block names
- Check character usage in block names
- Check display for port blocks
- Check whether block names appear below blocks
- Check for prohibited blocks in discrete controllers
- Check for prohibited sink blocks
- Check orientation of Subsystem blocks

To configure the checks that you want to run, in the model window, select **Analysis** > **Model Advisor** > **Configure Advisor Edit-Time Checks**.

For more information, see Check for Compliance in Model and Subsystems.

Clone and Variant Detection: Identify modeling patterns to refactor and simplify model

In R2016b, use the Model Transformer tool to refactor a model to reuse subsystem clones and implement variants. Clones are subsystems that have identical structures and parameter settings.

- Check for modeling patterns that qualify for transformation into Variant Source or Variant Subsystem blocks.

- Create a model that transforms modeling patterns into Variant Source or Variant Subsystem blocks.
- Check for clones.
- Create a model that replaces clones with links to subsystem blocks in a library.

Replacing clones with links to library blocks enables component reuse. If you have Simulink Coder or Embedded Coder software, you can generate reusable code for library subsystems.

Using variant blocks in a model helps reusablility of the model for different conditional expressions called variant choices. If you have Embedded Coder software, you can generate code containing preprocessor conditionals that control the activation of each variant choice.

See Check subsystem names.

MISRA Compliance: Run MISRA checks available in Simulink Verification and Validation

If you have the Simulink Verification and Validation software, you can run the Model Advisor checks in the **By Task** > **Modeling Guidelines for MISRA C:2012** folder. Check your models against patterns and settings that can lead to MISRA® violations. Previously, you only ran these checks with the Embedded Coder software.

Improved Model Advisor start and check execution speed

In R2016b, Model Advisor provides these performance improvements:

- Model Advisor start speed
- Check execution speed

Compatibility Considerations

Previously, to see changes to your custom Model Advisor checks, you restarted your MATLAB session or entered `sl_refresh_customizations` at the command line. In R2016b, to see your changes, at the command line, enter Advisor.Manager.refresh_customizations or take the following steps:

1. From the Model Editor, select **Analysis** > **Model Advisor** > **Preferences**.

2 In the Model Advisor Preferences window, click the **Update check information cache** button.

Cache metric data with `slmetric.Engine` class

In R2016b, when you use the slmetric.Engine.execute method to collect metric data:

- The generated metrics are persistent in the simulation cache folder. Future instantiations of the `slmetric.Engine` object for the same model can access the metrics without collecting them.
- Subsequent calls to the slmetric.Engine.execute method do not collect metric data unless the model or the metric algorithm `Version` property has changed.

Processor-in-the-Loop Code Coverage: Measure code coverage in Processor-in-the-Loop

If you have Embedded Coder software, Simulink Verification and Validation can measure coverage for code generated from models in PIL mode. This is similar to measuring coverage in SIL mode. For more information, see Code Coverage for Models in Software-in-the-Loop (SIL) Mode and Processor-in-the-Loop (PIL) Mode.

MCDC Model Coverage: Record MCDC coverage for cascaded Simulink logic blocks

When recording coverage for models that include Simulink logic blocks, each cascade of logic blocks receives a singular MCDC result for the Boolean expression represented by the cascade if Treat Simulink logic blocks as short-circuited is enabled. For more information on cascaded Simulink logic blocks, see Analyzing MCDC for Cascaded Logic Blocks, Logical Operator Cascade Patterns, and Modified Condition and Decision Coverage in Simulink Design Verifier.

Display Coverage Results: Access, manage, and accumulate model coverage results with Coverage Results Explorer

Use the Coverage Results Explorer dialog box to access, manage, and accumulate coverage data post-analysis. For instructions on how to use the Coverage Results Explorer, see Access, Manage, and Accumulate Coverage Results.

Integrated Coverage Configuration Settings: Access coverage settings integrated within the Configuration Parameters dialog box

Access Model Coverage settings from the Coverage pane in the Configuration Parameters dialog box. The settings interface is redesigned for typical workflows. For more information, see Specify Model Coverage Options.

Harness Requirements Linking: Link requirements to model objects in externally stored test harnesses

You can now associate requirements with model objects in externally stored test harnesses created with Simulink Test™.

Requirements Traceability: RMI data storage is now external by default

As you add RMI links to a Simulink model, a separate file is created to hold the RMI data. Adding RMI links will therefore not result in a dirty model. By default, the name and location of the new file matches the name and location of the Simulink model. You can specify other names and locations via the context menu.

R2016a

Version: 3.11

New Features

Bug Fixes

Model Metric APIs: Programmatically run and create metrics

The DO-331, IEC 61508, IEC 62304, ISO 26262, and EN 50128 standards recommend that your model and code comply with size, complexity, and readability requirements. To help you assess your model, use the model metrics available with Simulink Verification and Validation. You can also use APIs to create your own model metrics, compute metrics, and export metric data.

Action	Use	See
Create your own model metrics.	Model metric customization APIs	Create Model Metrics by Using APIs New classes: • slmetric.metric.Metric • Advisor.component.Component • Advisor.component.Types
Programmatically assess the size, complexity, and readability of your model.	Model metric APIs	Collect Model Metrics Programmatically New classes: • slmetric.Engine • slmetric.metric.Result • slmetric.metric.ResultCollection
Interactively assess the size, complexity, and readability of your model. Previously available in R2015b.	Model Advisor metric checks	Run Checks for Model Metrics

Model Advisor Checks for Medical Standard IEC 62304: Check models to comply with medical standards

To check for model compliance to medical standard IEC 62304, run the Model Advisor checks in the **By Task > Modeling Standards for IEC 62304** folder.

Model Advisor Standards Checks: Check models for naming standard compliance

The Model Advisor includes standard checks that you can use to verify that your model complies with high-integrity modeling guideline hisl_0032: Model object names.

You can use the Model Advisor Configuration Editor to specify the naming standard and model objects that the checks flag.

DO-178C/DO-331

The Model Advisor **By Task** folder contains the **Modeling Standards for DO-178C/DO-331** check, Check model object names.

IEC 61508, IEC 62304, ISO 26262, and EN 50128

The Model Advisor **By Task** folder contains **Modeling Standards for IEC 61508**, **Modeling Standards for IEC 62304**, **Modeling Standards for ISO 26262**, and **Modeling Standards for EN 50128** check, Check model object names.

Model Advisor Check Parameterization: Control check behavior by using input parameters

In the Model Advisor Configuration Editor, to specify conditions that control check behavior, select input parameters.

Modeling Standard	Check	Input Parameters
DO-178C DO-331	Check model object names	To specify the check naming standard and model objects, select `MAAB`, `Custom`, or `None` for each of the following model objects: • Blocks • Signals • Parameters • Busses • Stateflow® objects
IEC 61508, IEC 62304, ISO 26262, and EN 50128	Check model object names	

Modeling Standard	Check	Input Parameters
		If you select `Custom`, you can enter your own regular expression for flagging the specified model object names.
MAAB	Check character usage in block names	To specify the check **Naming standard** for blocks, select `MAAB 3.0` or `Custom`. If you select `Custom`, you can enter your own regular expression for flagging the block names.
	Check port block names	To specify the check **Naming standard** for Inport and Outport blocks, select `MAAB 3.0` or `Custom`. If you select `Custom`, you can enter your own regular expression for flagging the Inport and Outport block names.
	Check subsystem names	To specify the check **Naming standard** for subsystem names, select `MAAB 3.0` or `Custom`. If you select `Custom`, you can enter your own regular expression for flagging the subsystem block names.
	Check for mixing basic blocks and subsystems	In the **Input Parameters** pane, to: • Allow the blocks specified by MAAB, from **Standard**, select `MAAB 3.0`. The **Block type list** table provides the blocks that MAAB 3.0 allows at any model level. • To specify blocks to either allow or prohibit, from **Standard**, select `Custom`. In **Treat blocktype list as**, select `Allowed` or `Prohibited`. In the **Block type list** table, you can add or remove blocks. Previously, you specified only blocks and subsystems that the check allowed.

Modeling Standard	Check	Input Parameters
	Check for prohibited blocks in discrete controllers	In the **Input Parameters** pane, to: • Prohibit the blocks as specified in MAAB, from **Standard**, select `MAAB 3.0`. The **Block type list** table provides the blocks that MAAB 3.0 prohibits inside controllers. • To specify blocks to either allow or prohibit, from **Standard**, select `Custom`. In **Treat blocktype list as**, select `Allowed` or `Prohibited`. In the **Block type list** table, you can add or remove blocks. Previously, you specified only blocks and subsystems that the check prohibited.
	Check for prohibited sink blocks	In the **Input Parameters** pane, to: • Prohibit the blocks as specified by MAAB 3.0, from **Standard**, select `MAAB 3.0`. The **Block type list** table provides the sink blocks that MAAB 3.0 prohibits. • To specify blocks to either allow or prohibit, from **Standard**, select `Custom`. In **Treat blocktype list as**, select `Allowed` or `Prohibited`. In the **Block type list** table, you can add or remove blocks. Previously, you specified only blocks and subsystems that the check prohibited.

To specify the input parameters, use the Model Advisor Configuration Editor.

1. Open the Model Configuration Editor and navigate to the check.
2. In the **Input Parameter** pane, specify the parameters.
3. Click **Apply**.
4. Save the configuration. When you run the checks with this configuration, the checks flag the model conditions that you specified.

Code Coverage for Generated Code: Measure testability of generated code with coverage analysis

If you have Embedded Coder installed, Simulink Verification and Validation measures the coverage of code generated from models in SIL mode. For more information, see Coverage for Models in Software-in-the-Loop (SIL) Mode.

Requirements Linking: Trace requirements to objects in test harnesses and steps in Test Sequence blocks

If you have Simulink Test installed, you can associate requirements with objects in test harnesses and with steps in Test Sequence blocks in Simulink diagrams. For more information, see Link Test Cases to Requirements Documents.

Coverage Data Collection: Automatically save cumulated coverage data to file

When you record coverage for a model, Simulink Verification and Validation records the current coverage data and cumulated coverage data to a .cvdata file on the disk.

Model Coverage Analysis: Justification editor to identify acceptable instances of incomplete coverage from report

When you create a coverage filter for a construct, you choose between *Excluded* and *Justified* filtering modes. Simulink Verification and Validation includes justified constructs in the coverage report as a subset of the constructs that achieve coverage.

For more information, see:

- Filter Model Objects to Refine Coverage Results
- Creating and Using Coverage Filters

Coverage results for test files, test suites, and test cases in Simulink Test

If you have a Simulink Test license, you can aggregate coverage results for test cases, test suites, and test files using the Test Manager. You can include coverage in the Test Manager results report. See Collect Coverage in Tests.

R2015aSP1

Version: 3.9.1

Bug Fixes

R2015b

Version: 3.10

New Features

Bug Fixes

Compatibility Considerations

Model Advisor checks for size, cyclomatic complexity, and readability model metrics

To display model information and help you assess your model and code, you can run the count, complexity and readability model metrics checks in the applicable **By Task** > **Model Metrics** subfolder. The following table lists the model metric checks. For an example of running checks for model metrics, see "Collect Model Metrics Using the Model Advisor".

Subfolder	Model Metric Checks
Count Metrics	• "Simulink block metric" • "Subsystem metric" • "Library link metric" • "Effective lines of MATLAB code metric" • "Stateflow chart objects metric" • "Lines of code for Stateflow blocks metric" • "Subsystem depth metric"
Complexity Metrics	• "Cyclomatic complexity metric"
Readability Metrics	• "Nondescriptive block name metric" • "Data and structure layer separation metric"

Model Advisor reports in Adobe PDF and Microsoft Word formats

When the Model Advisor runs checks, by default, it generates an HTML report of check results. On Windows® platforms, you can now generate Model Advisor reports in Adobe® PDF and Microsoft® Word .docx formats. After you run a Model Advisor analysis, to generate a Model Advisor report in PDF or .docx format, in the right-pane of the Model Advisor window, select **Generate Report**. For more information, see "Generate Model Advisor Reports in Adobe PDF and Microsoft Word Formats".

Model Advisor check support for JMAAB Guidelines, Version 4.0

The MathWorks® Automotive Advisory Board (MAAB) checks support compliance with MAAB Control Algorithm Modeling Guidelines, Version 3.0. A subset of the MAAB checks support compliance with the Japan MBD Automotive Advisory Board (JMAAB) Control Algorithm Modeling Guidelines, Version 4.0. For a summary of the MAAB check support for the MAAB and JMAAB Guidelines, see “Model Checks for MathWorks Automotive Advisory Board (MAAB) Guideline Compliance”.

To download the JMAAB Control Algorithm Modeling Guidelines, Version 4.0, see the JMAAB website.

Block execution recorded during model coverage analysis

Simulink Verification and Validation can perform several types of coverage analysis. The model coverage tool now records execution coverage for each block in a model. Block execution coverage determines whether or not a block is executed during model simulation. For more information, see “Types of Model Coverage”.

.

Manage model coverage parameters within a model configuration set

A model configuration set stores model coverage parameters for a model when you load or create a model. This feature allows for the storage of model coverage configuration parameters that you can reuse for multiple models or model variants. For more information, see “About Model Configurations” (Simulink).

Model Advisor support for MISRA C:2012

Model Advisor checks support compliance with MISRA C:2012. Previously, Model Advisor checks supported compliance with MISRA C:2004. The check documentation references the applicable MISRA C:2012 directive or rule. For information about MISRA C® versions and updates, see “MISRA C Guidelines” (Embedded Coder) in the Embedded Coder documentation.

For check documentation, see:

- “Model Checks for DO-178C/DO-331 Standard Compliance”
- “Model Checks for IEC 61508, IEC 62304, ISO 26262, and EN 50128 Standard Compliance”
- “Model Checks for MathWorks Automotive Advisory Board (MAAB) Guideline Compliance”

API to run Model Advisor across model hierarchy

Use instances of Advisor.Application to run Model Advisor checks across a model hierarchy. You can use `Advisor.Application` to:

- Run checks on referenced models.
- Select model components for Model Advisor analysis.
- Select checks to run during Model Advisor analysis.

If you have a large model with subsystems and model references, consider using `Advisor.Application`. If you want to leverage parallel processing, use `ModelAdvisor.run` to run Model Advisor checks on your model.

Model Advisor check exclusions saved to model SLX file

When you exclude blocks and charts from Model Advisor checks, you can save the exclusion information to the model SLX file. In the Model Advisor Exclusion Editor dialog box, select **Store exclusions in model file**. When you open a model SLX file, the model contains the Model Advisor check exclusions. Previously, the exclusion information was stored only in Model Advisor exclusion files `<`*`model_name`*`>_exclusions.xml` attached to a model. For more information, see “Limit Model Checks”.

Compatibility Considerations

You can still use Model Advisor exclusion files `<`*`model_name`*`>_exclusions.xml` to store exclusion information for your Model Advisor analysis. When you open a model that has an attached exclusion file, to help you manage exclusions, consider saving the exclusions in the model file.

Model Advisor check input parameterization

You can use input parameters to control the blocks that are flagged when you run the following checks.

- “Check for mixing basic blocks and subsystems”
- “Check for prohibited blocks in discrete controllers”
- “Check for prohibited sink blocks”

To change the list of blocks that the checks flag, use the Model Advisor Configuration Editor.

1. Open the Model Configuration Editor and navigate to the check.
2. In the **Input Parameter** pane, add or remove blocks.
3. Click **Apply**.
4. Save the configuration. When you run the checks using this configuration, the checks flag the blocks that you specified.

Functionality Being Removed or Changed

Functionality	Result	Use Instead	Compatibility Considerations
`rmidata.default`	Error	`rmipref`	Replace all instances of `rmidata.default` with `rmipref`.

R2015a

Version: 3.9

New Features

Bug Fixes

Coverage measurement for C S-functions

Model coverage is supported for C/C++ S-Functions. The coverage results for S-Function blocks can be viewed in the same report as the rest of the model. For each S-Function block, the report links to a detailed coverage report for the C/C++ code in the block.

To generate coverage report for S-Function blocks:

1. When creating S-Functions, enable support for coverage.
2. When generating coverage report, enable support for S-Functions.

The following coverage metrics are reported for S-Functions:

- Cyclomatic Complexity
- Condition Coverage (CC)
- Decision Coverage (DC)
- MCDC Coverage
- Percentage of statements covered

For more information, see:

- Model Coverage for C and C++ S-Functions
- View Coverage Results for C/C++ Code in S-Function Blocks

Model Advisor checks for MATLAB functions

The Model Advisor includes checks for MATLAB functions to comply with modeling standards. Previously, the checks verified that only MATLAB Function blocks complied with modeling standards.

IEC 61508, ISO 26262, and EN 50128

The Model Advisor **By Task** folder contains **Modeling Standards for IEC 61508**, **Modeling Standards for ISO 26262**, and **Modeling Standards for EN 50128** checks:

- Check for MATLAB Function interfaces with inherited properties replaces **Check for MATLAB Function block interfaces with inherited properties**.
- Check MATLAB Function metrics replaces **Check MATLAB Function block metrics**.

DO-178C/DO-331

The Model Advisor **By Task** folder contains **Modeling Standards for DO-178C/DO-331** checks:

- Check for MATLAB Function interfaces with inherited properties replaces **Check for MATLAB Function block interfaces with inherited properties**.
- Check MATLAB Function metrics replaces **Check MATLAB Function block metrics**.

MathWorks Automotive Advisory Board

The Model Advisor **By Task > Modeling Standards for MAAB > MATLAB Functions** folder contains new checks:

- Check input and output settings of MATLAB Functions replaces **Check input and output settings of MATLAB Function blocks**.
- Check MATLAB Function metrics replaces **Check MATLAB Function block metrics**.

R2014b

Version: 3.8

New Features

Bug Fixes

Model coverage for relational boundary values

The model coverage tool introduces a relational boundary coverage metric.

Relational boundary coverage examines blocks that have an explicit or implicit relational operation. The coverage also examines Stateflow transitions and MATLAB function blocks that contain a relational operation.

For these blocks, the metric records whether a simulation tests the relational operation with:

- Equal operand values. This part of relational boundary coverage applies only if both operands are integers or fixed-point numbers.
- Operand values that differ by a certain tolerance. This part of relational boundary coverage applies to all operands. For integer and fixed-point operands, the tolerance is fixed. For floating-point operands, you can use either a predefined tolerance or specify your own tolerance value.

For more information, see Relational Boundary Coverage.

Modeling standard check support for charts that use MATLAB as the action language

The following Model Advisor checks now support charts that use MATLAB as the action language.

In the **By Task** > **Modeling Standards for IEC 61508**, **Modeling Standards for ISO 26262**, and **Modeling Standards for EN 50128** folders:

- Check usage of Stateflow constructs

In the **By Task** > **Modeling Standards for MAAB** > **Stateflow** folder:

- Check transition actions in Stateflow charts
- Check for unary minus operations on unsigned integers in Stateflow charts
- Check for equality operations between floating-point expressions in Stateflow charts
- Check entry formatting in State blocks in Stateflow charts
- Check return value assignments of graphical functions in Stateflow charts
- Check usage of return values from a graphical function in Stateflow charts

- Check for event broadcasts in Stateflow charts
- Check for comparison operations in Stateflow charts
- Check usage of exclusive and default states in state machines

Requirements linking for Simulink data dictionary entries

You can create requirements traceability links for entries in Simulink data dictionaries. In the Model Explorer, right-click a data dictionary entry. From the Requirements Traceability context menu, use selection-based linking shortcuts or the Requirements Link Editor.

Capture and display of requirements details in model highlighting and Web view

Requirements traceability highlighting opens a separate Requirements Details window that shows content from linked documents in Microsoft Word, Microsoft Excel®, and IBM Rational DOORS. To open the Requirements Details window, in the Simulink Editor, select **Analysis** > **Requirements Traceability** > **Highlight Model**. Select a model object with requirements links to see content from linked requirements documents in the Requirements Details window.

If you have Simulink Report Generator™ installed, you can include requirements information in a Web view. In the Simulink Editor, select **File** > **Export Model to** > **Web** In the Web View dialog box, open the **Optional Views** tab and select **Include Requirements view**. Alternatively, in the Simulink Editor, select **Analysis** > **Requirements Traceability** > **Generate Web View**.

To view requirements information in a Web view, click a highlighted block (for example, orange highlighted blocks have requirements information). The information for that block appears in the **Informer** pane underneath the model.

R2014a

Web view option for coverage results

You can export a Model Coverage Web View for your model. A Web View is an interactive rendition of a model that you can view in a Web browser. A Model Coverage Web View includes model coverage highlighting and analysis information from the Coverage Display Window, as described in View Coverage Results in a Model. For more information, see Export Model Coverage Web View.

Model Advisor check authoring for model configuration parameters

You can create a configuration parameter check by specifying check behavior in an XML data file. To register the check, you use an sl_customization file. Previously, only DO-178C/DO-331 checks verified configuration parameter checks. To create the configuration parameter check, see Create Check for Model Configuration Parameters.

Condition coverage for Relational Operator blocks

The model coverage tool records condition coverage for Relational Operator, Compare to Constant, and Compare to Zero blocks. For more information, see Model Objects That Receive Coverage.

Model Advisor checks for MATLAB Code Analyzer messages

DO-178C/DO-331

The Model Advisor **By Task** folder contains these **Modeling Standards for DO-178C/ DO-331** checks:

- Check MATLAB Code Analyzer messages
- Check MATLAB code for global variables

IEC 61508, ISO 26262, and EN 50128

The Model Advisor **By Task** folder contains these **Modeling Standards for IEC 61508**, **Modeling Standards for ISO 26262**, and **Modeling Standards for EN 50128** checks:

- Check MATLAB Code Analyzer messages
- Check MATLAB code for global variables

MathWorks Automotive Advisory Board

The Model Advisor **By Task > Modeling Standards for MAAB > MATLAB Functions** folder contains this new check:

- Check MATLAB code for global variables

Japanese language localization support

Simulink Verification and Validation is available in Japanese for Japanese localized systems. For more information, see Internationalization.

Enhanced modeling standards checks for display attributes and Math Operations blocks

MathWorks Automotive Advisory Board check for display attributes

Check the display attributes of block names checks the display attributes of all blocks to verify compliance with MathWorks Automotive Advisory Board guideline jc_0061: Display of block names. Previously, the check did not verify compliance of subsystem block names. You can find **Check the display attributes of block names** in the Model Advisor **By Task > Modeling Standards for MAAB > Simulink** folder.

IEC 61508, ISO 26262, and EN 50128 Check usage of Math Operations blocks

IEC 61508, ISO 26262, and EN 50128 Check usage of Math Operations blocks verifies the use of the following Math Operations blocks:

- Abs
- Assignment
- Gain

Previously, the check verified the use of only Abs and Gain blocks. **Check usage of Math Operations blocks** is available in the **By Task >Modeling Standards for IEC 61508**, **Modeling Standards for ISO 26262**, and **Modeling Standards for EN 50128** folders.

DO-178C/DO-331 Check usage of Math Operations blocks

DO-178C/DO-331 Check usage of Math Operations blocks verifies the use of the following Math Operation blocks:

- Abs
- Assignment
- Gain
- Math Function
  - Natural logarithm
  - Common (base 10) logarithm
  - Remainder after division
  - Reciprocal

Previously, the check verified the use of only Abs and Gain blocks. The DO-178C/DO-331 **Check usage of Math blocks** verified the use of the Math Function blocks. **Check usage of Math blocks** is no longer available in the **By Task > Modeling Standards for DO-178C/DO-331** folder.

Check usage of Math Operations blocks is available in the Model Advisor **By Task > Modeling Standards for DO-178C/DO-331** folder.

R2013b

Version: 3.6

New Features

Bug Fixes

Compatibility Considerations

Requirements traceability for MATLAB code

When you store requirements traceability data externally, you can create traceability links to and from specific lines of MATLAB code. This capability allows you to include finer granularity in your traceability links. For example, instead of linking between a requirement and a MATLAB Function block, you can link between a requirement and individual lines or sections of code inside a MATLAB Function block.

For more information, see MATLAB Code Traceability.

Reset option for cumulative model coverage results

If you have enabled cumulative model coverage data collection and have existing cumulative coverage data, you can reset that data between simulations. In the Coverage Settings dialog box, on the **Results** tab, click **Clear data**. For more information, see Specify Model Coverage Options.

EN 50128 modeling standard checks for railway applications

The Model Advisor includes model checks for compliance to the EN 50128 standard. The checks are available in the **By Task > Modeling Standards for EN 50128** folder.

Model Advisor checks for MATLAB Function blocks and input/output range specifications

MATLAB Function blocks

The Model Advisor includes checks for MATLAB Function blocks to comply with modeling standards.

IEC 61508, ISO 26262, and EN 50128

The Model Advisor **By Task** folder contains **Modeling Standards for IEC 61508**, **Modeling Standards for ISO 26262**, and **Modeling Standards for EN 50128** checks:

- Check for MATLAB Function block interfaces with inherited properties
- Check MATLAB Function block metrics

DO-178C/DO-331

The Model Advisor **By Task** folder contains **Modeling Standards for DO-178C/DO-331** checks:

- Check for MATLAB Function block interfaces with inherited properties
- Check MATLAB Function block metrics

MathWorks Automotive Advisory Board

The Model Advisor **By Task > Modeling Standards for MAAB > MATLAB Functions** folder contains new checks:

- Check input and output settings of MATLAB Function blocks
- Check MATLAB Function block metrics

Inport and Outport range specifications

The Model Advisor includes checks for MATLAB Function blocks to comply with modeling standards.

IEC 61508, ISO 26262, and EN 50128

The Model Advisor **By Task** folder contains **Modeling Standards for IEC 61508**, **Modeling Standards for ISO 26262**, and **Modeling Standards for EN 50128** checks:

- Check for root Inports with missing range definitions
- Check for root Outports with missing range definitions

`ModelAdvisor.run` option for parallel processing

If you have a Parallel Computing Toolbox™ license, you can check multiple systems in parallel by using the ModelAdvisor.run function with `'ParallelMode’` set to `'On'`. By default, `‘ParallelMode’` is set to `’Off’`.

Compatibility Considerations

When you use ModelAdvisor.run with `'ParallelMode’` set to `'On'`, MATLAB automatically creates a parallel pool. Previously, when using `ModelAdvisor.run` to check multiple systems in parallel, you had to open a parallel pool.

R2013a

Version: 3.5

New Features

Bug Fixes

Model coverage for saturation on integer overflow in Stateflow, MATLAB Function blocks

You can use the model coverage tool to analyze saturation on integer overflow in Stateflow charts and MATLAB Function blocks. During simulation, the software detects the number of times saturation on integer overflow occurs in a Stateflow chart or MATLAB function.

Saturation on integer overflow coverage for Simulink blocks has been supported since R2012a.

To collect saturation on integer overflow coverage for your model, in the Simulink Editor, select **Analysis** > **Coverage** > **Settings**. In the Coverage Settings dialog box, in the Coverage metrics group, select **Saturation on integer overflow**.

`rmipref` command for managing requirements traceability preferences

The rmipref function provides a command-line interface that enables you to manage requirements traceability preferences.

For more information on requirements traceability preferences, see The Requirements Settings Dialog Box.

Inclusion of DOORS module-prefix in requirements traceability location identifiers

Unique target identifiers for IBM Rational DOORS requirements links now include DOORS module prefixes, if they are present.

Model Advisor navigation between Upgrade Advisor, Performance Advisor, and Code Generation Advisor

In the Model Advisor window, you can select:

- **Code Generation Advisor** to help configure your model to meet code generation objectives.
- **Upgrade Advisor** to help upgrade models.
- **Performance Advisor** to help improve the simulation performance of your model.

New Model Advisor Checks for Modeling Standards

The Simulink Model Advisor includes new modeling standards checks.

IEC 61508 and ISO 26262 Checks

The Model Advisor **By Task** >**Modeling Standards for IEC 61508** and **Modeling Standards for ISO 26262** folders contain a new check to replace **Check for questionable constructs**:

- Check for blocks not recommended for C/C++ production code deployment

DO-178C/DO-331 Checks

The Model Advisor **By Task** folder contains a new **Modeling Standards for DO-178C/ DO-331** check to replace **Identify questionable blocks within the specified system**:

- Check for blocks not recommended for C/C++ production code deployment

R2012b

Version: 3.4

New Features

Bug Fixes

Compatibility Considerations

Custom labels for requirements links to IBM Rational DOORS

You can specify custom templates for labels of your model requirements links to IBM Rational DOORS. With these templates, in your model, you can include DOORS object property information that you need for your project. The Simulink Verification and Validation software uses requirements links labels in context menus, generated reports, code comments, and consistency checks.

Using the rmi command, you can specify a custom template for new DOORS requirements links. You can include user-defined DOORS attributes in the template. For more information, at the MATLAB command prompt, enter:

```
help rmi
```

Under the heading

```
Custom labels for DOORS links
```

the output shows detailed examples for available template settings.

Compatibility Considerations

If you specify a custom template for labels that differs from the default template and your model has existing requirements links to DOORS, requirements consistency checking reports inconsistencies for these links. You can adjust labels for existing DOORS requirements links using the rmi command.

Requirements linking for Simulink annotations

You can create requirements links to and from `Simulink.Annotation` objects using the Requirements Management Interface (RMI). Annotations are free-floating text boxes that you can place inside a Simulink model. You can use them to describe requirements for the design. You can then use RMI linking to associate annotations with other Simulink objects or external requirements documents.

Requirements linking for Simulink annotations is enabled only if you configure your model to store requirements data externally. To specify external storage of requirements data for a model, in the Requirements Settings dialog box under **Storage** > **Default storage location for requirements links data**, select **Store externally (in a separate *.req file)**.

For more information, see Requirements Linking with Simulink Annotations.

Compatibility Considerations

If you have a model that already contains requirements links using the internal storage setting, you cannot use requirements linking for Simulink annotations until you convert that model to use external storage for RMI data. To convert the model, move all existing requirements links in it to an external `.req` file by selecting **Analysis** > **Requirements** > **Move to File**.

Coverage filtering for Stateflow events

You can exclude Stateflow events from recording coverage during Simulink Verification and Validation model coverage collection. For more information, see Filter a Stateflow Event.

Model Advisor dashboard, providing efficient model guideline checking

The Model Advisor dashboard provides a way for you to efficiently check that your model complies with modeling guidelines. You can use the Model Advisor dashboard to run a set of checks on your model without opening the Model Advisor window. For more information, see Overview of the Model Advisor Dashboard.

Model Advisor checks for library models and the DO-178C/DO-331 standard

Library Models

There are Model Advisor checks available to verify the syntax of library models. When you use the Model Advisor to check a library model, the Model Advisor window indicates (~) checks that do not check libraries. To determine if you can run the check on library models, you can also refer to the check documentation, “Capabilities and Limitations”. You cannot use checks that require model compilation. You can use the ModelAdvisor.Check.SupportLibrary API to create custom checks for library models.

DO-178C/DO-331 Standard

The Model Advisor also includes model checks for compliance to the DO-178C and DO-331 standards. The new checks are available in the Model Advisor folders:

- **By Task > Modeling Standards for DO-178C/DO-331**

- **By Product > Simulink Verification and Validation > Modeling Standards > Modeling DO-178C/DO-331 Checks**

For information on the DO-178C Software Considerations in Airborne Systems and Equipment Certification and related standards, see Radio Technical Commission for Aeronautics (RTCA) .

Former DO-178B checks are available in the DO-178C/DO-331 Model Advisor folders. The DO-178C/DO-331 checks replace the DO-178B checks.

HTTP navigation from generated requirements report to requirements and Simulink objects

You can set the Requirements Management Interface (RMI) to generate requirements reports that support navigation from system browsers to requirements documents and model objects. You can view and navigate from your generated reports in system browsers as long as the MATLAB internal HTTP server is active on your local host. If you convert your generated report to a PDF file or Microsoft Word document, the navigation links remain valid. For more information, see Customize Requirements Report Using the RMI Settings and Link from External Applications.

R2012a

Version: 3.3

New Features

Bug Fixes

Compatibility Considerations

Coverage Metric for Blocks with Saturate on Integer Overflow Parameter

In R2012a, the Simulink Verification and Validation model coverage tool provides coverage for blocks when you select the **Saturate on Integer Overflow Coverage** parameter option. The software reports the number of times a block saturates on integer overflow during simulation.

To enable this feature, in the Coverage Settings dialog box, on the **Coverage** tab, select the **Saturate on Integer Overflow** option.

Coverage Support for Simulink Design Verifier Functions Inside Stateflow Charts

Simulink Verification and Validation now records coverage for the following Simulink Design Verifier functions inside Stateflow charts:

- sldv.assume
- sldv.condition
- sldv.prove
- sldv.test

For more information, see Simulink Design Verifier Coverage.

Cumulative Coverage

Simulink Verification and Validation now provides cumulative coverage for identically configured multiple instances of:

- Reusable subsystems
- Stateflow constructs

For more information, see Creating Cumulative Coverage for Reusable Subsystems and Stateflow Constructs.

Requirements Linking Between Simulink Objects

You can now create requirements links between Simulink objects using selection-based linking. This capability allows you to create requirements links between models, enhancing traceability.

To create a requirement link from one model object to another model object:

1 Right-click the link destination model object and select **Requirements > Select for linking with Simulink**.
2 Right-click the link source model object and select **Requirements > Add link to selected object**.
3 Right-click the link source model object again and select **Requirements**. The new link appears at the top of the **Requirements** submenu.

For more information, see Links Between Models and Requirements.

URL-Based Navigation From External Requirements Documents to Simulink Objects

In R2012a, the Simulink Verification and Validation tool provides URL-based navigation from external requirements documents to Simulink objects. Previously, the software used ActiveX® controls to navigate from Microsoft Office documents to Simulink objects. Now, by default, navigation links from Microsoft Office documents are hyperlinks, allowing for export to HTML or PDF files, and manual copy-paste into other document formats. For more information, see The Requirement Settings Dialog Box.

Compatibility Considerations

Compatibility Considerations

Simulink Verification and Validation software continues to recognize navigation links created with ActiveX controls. Additionally, you can create navigation links with ActiveX by selecting **Use ActiveX buttons in Word and Excel (backward compatibility)** in the Requirements Settings dialog box.

Model Advisor Exclusions

You can now exclude model blocks from selected Model Advisor checks, saving time during model development and verification. When a Model Advisor analysis is complete, you can use Model Advisor highlighting to indicate the analysis results for individual Model Advisor checks in the model window. You can exclude blocks from selected checks by right-clicking blocks and selecting exclusions options. For more information, see Limiting the Scope of a Model Advisor Analysis.

MathWorks Automotive Advisory Board Checks

In R2012a, the Model Advisor **By Task > Modeling Standards for MAAB > Stateflow** folder no longer includes:

- **Check for annotations with drop shadows**.
- **Check for tunable parameters in Stateflow charts**.
- **Check whether Stateflow events are defined at the chart level or below**. This check is relevant only for software versions previous to release R2009b.

For more information, see MathWorks Automotive Advisory Board Checks in the Simulink Verification and Validation documentation.

R2011b

Version: 3.2

New Features

Bug Fixes

Compatibility Considerations

Condition and MCDC Coverage for If Block Conditions

The If block now receives condition and MCDC coverage on any `if` conditions or `elseif` conditions that contain logical expressions with multiple conditions. Previously, the If block only received decision coverage.

The If block receives coverage as follows:

- The software measures decision coverage for the `if` condition and all `elseif` conditions.
- If the `if` condition or any of the `elseif` conditions contain a logical expression with multiple conditions, such as `u1 & u2 & u3`, the software also measures condition and MCDC coverage for each condition in the expression, for example, `u1`, `u2`, and `u3` .

Capability to Store Requirements Information Outside the Model File

The Requirements Management Interface (RMI) now provides an option to store requirements links information in an external file. The default name for this file is *`model_name`*`.req`. External storage allows you to:

- Associate information about links to external requirements documents without modifying the timestamp and version of your model (`.mdl`) file.
- Use different sets of requirements links with the same model by loading links from different files.

To enable external storage of requirements links, in the Requirements Settings dialog box, on the **Storage** tab, select **Store externally (in a separate .req file)**. After setting this option, all models for which you create new requirements links will use external storage.

The RMI provides several new functions that allow you to manage the links in external files:

- rmidata.map — Update link information in an external file.
- rmidata.default — Set the option for storing requirements link information (`'internal'` or `'external'`).
- rmidata.export — Move link information for a model to an external file.

Selection-Based Linking for Custom Requirements Documents

If you link to requirements in custom requirements documents, you can now define link type properties that provide selection-based linking capabilities for these documents. Selection-based linking will be available when you:

- Right-click a model object and select Requirements in the context menu.
- Select **Requirements** > **Edit/Add Links** and create new requirements using the Requirements dialog box.

For more information, see Custom Types of Requirements Documents.

Include Requirements Document Data in the Report

The Requirements Management Interface (RMI) can now include details from requirements text in the requirements report, depending on the type of requirements document:

- Microsoft Word — Include entire paragraph or subsection associated with each linked requirement.
- Microsoft Excel — If the requirement is a single cell in a workbook, include the content from all the preceding cells in that row as a table. If the requirements are in a cell range, include all the content from the cells in that range as a table.
- IBM Rational DOORS — Include the **Object Text** and **Object Heading** attributes for the linked requirements object, and other attributes as specified by the user.

  You can specify which DOORS attributes to include or omit from the report using the new RptgenRMI.doorsAttribs function.

Before you generate the requirements report, in the Requirements Settings dialog box, on the **Report** tab, select **Include details from linked documents**.

Microsoft Excel Support for Consistency Checking of Simulink References in Requirements Documents

The rmi function has a `'checkdoc'` option that checks links from requirements documents to Simulink model objects. This option now checks Microsoft Excel requirements documents, in addition to Microsoft Word documents and IBM Rational DOORS databases.

Requirements Report Default File Names Based on Model Names

When the Requirements Management Interface (RMI) generates an HTML report file, it now uses the model name in the file name. The default name for a requirements report is *`model_name`*`_report.html`.

Changed Format for slvnvruntest and slvnvruncgvtest Output

The output format for slvnvruntest and slvnvruncgvtest 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_slvnvruncgvtest`	Simulation time
`xout_slvnvruncgvtest`	State data
`yout_slvnvruncgvtest`	Output signal data
`logsout_slvnvruncgvtest`	Signal logging data for: • Signals that are connected to outports • Signals that are configured for logging on the model

Compatibility Considerations

Compatibility Considerations

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

```
runOpts = slvnvruntestopts;
runOpts.outputFormat = 'TimeSeries';
slvnvruntest(model_name, sldvData, runOpts);
```

or

```
runOpts = slvnvruntestopts;
runOpts.outputFormat = 'StructureWithTime';
slvnvruntest(model_name, data, runOpts);
```

Conversion of Error and Warning Message Identifiers

For R2011b, error and warning message identifiers have changed in Simulink Verification and Validation.

Compatibility Considerations

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 `SLVNV:simcoverage:InvalidFirstArgument` identifier has changed to `SLVNV:simcoverage:complexityinfo:InvalidFirstArgument`. If your code checks for `SLVNV:simcoverage:InvalidFirstArgument`, you must update it to check for `SLVNV:simcoverage:complexityinfo:InvalidFirstArgument` 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.

API for Creating Procedural-Based Model Advisor Configurations

In R2011b, there is a new API you can use to create a procedural-based Model Advisor configuration. A procedural-based configuration allows you to specify the order in which you make changes to your model. You do this by organizing checks into procedures using the procedures API. A check in a procedure does not run until the previous check passes. A procedural-based configuration runs until a check fails, requiring you to modify the model to pass the check and proceed on to the next check. For more information, see Creating Procedural-Based Model Advisor Configurations.

New Model Advisor Checks

The Simulink Model Advisor includes new modeling standards checks.

ISO 26262 Checks

The Model Advisor **By Task** folder contains a new **Modeling Standards for ISO 26262** check:

- Check for inconsistent vector indexing methods

IEC 61508 Checks

The Model Advisor **By Task** folder contains a new **Modeling Standards for IEC 61508** check:

- Check for inconsistent vector indexing methods

DO-178B Checks

The Model Advisor **By Task** folder contains a new **Modeling Standards for DO-178B** check:

- Check for inconsistent vector indexing methods

Check Input Parameters in Model Advisor Report

In R2011b, the Model Advisor report now displays the check input parameters. For more information, see Viewing and Saving Model Advisor Reports in the Simulink documentation.

R2011a

New Capability to Filter Model Objects from Coverage Collection

When you simulate your Simulink model, coverage filtering allows you to filter certain model objects from recording coverage. The objects that you specify to exclude are stored in an external file.

- Individual Simulink blocks that receive coverage
- All blocks of a certain type
- Subsystems and all their contents
- Stateflow charts, states, transitions, and temporal events
- Individual library-linked blocks or Stateflow charts
- All reference blocks from a specific block library

Improved RMI Support for Reference Blocks

The Requirements Management Interface (RMI) provides improved support for managing requirements links on reference blocks.

- The new **Library Block Requirements** menu option allows you to view and navigate to requirements associated only with the library block. These links are available to all models that have reference blocks linked to library blocks that themselves have requirements links.
- When you create links from reference blocks to requirements, those links are only associated with the reference block in the current model. You cannot push the requirements links associated with reference blocks back to the library.

New Function to Merge Logged Component Data

The slvnvmergedata function allows you to combine logged test cases and counterexamples into a single structure. You can use logged data created by running slvnvlogsignals, or by running a Simulink Design Verifier analysis.

New and Enhanced Model Advisor Checks

The Simulink Model Advisor includes new and updated modeling standards checks.

ISO 26262 Checks

In the Model Advisor **By Task** folder, there is a new **Modeling Standards for ISO 26262** subfolder.

The subfolder includes the following checks:

- Display model metrics and complexity report
- Check for fully defined interface
- Check for unconnected objects
- Check for questionable constructs
- Check state machine type of Stateflow charts
- Check for model objects that do not link to requirements
- Display configuration management data
- Check usage of Math Operations blocks
- Check usage of Stateflow constructs
- Check usage of Signal Routing blocks
- Check usage of Logic and Bit Operations blocks
- Check usage of Ports and Subsystems blocks

For more information, see the IEC 61508 and ISO 26262 Checks in the Simulink Verification and Validation documentation.

IEC 61508 Checks

Previously, the **IEC 61508 Checks** included the check:

- **Check usage of Simulink constructs**

This check is replaced by the following new checks:

- Check usage of Math Operations blocks
- Check usage of Signal Routing blocks
- Check usage of Logic and Bit Operations blocks
- Check usage of Ports and Subsystems blocks

For more information, see the IEC 61508 and ISO 26262 Checks in the Simulink Verification and Validation documentation.

DO-178B Checks

Previously, the **DO-178B Checks** included the following checks:

- **Check for proper usage of For Iterator blocks**
- **Check for proper usage of While Iterator blocks**
- **Check for proper usage of Relational Operator blocks**
- **Check for proper usage of blocks that compute absolute values**

These previously available checks are replaced by the following new checks:

- Check usage of Math Operations blocks
- Check usage of Signal Routing blocks
- Check usage of Logic and Bit Operations blocks
- Check usage of Ports and Subsystems blocks

Also available are the new checks:

- Check usage of lookup table blocks
- Check Stateflow charts for uniquely defined data objects

For more information, see the DO-178B Checks in the Simulink Verification and Validation documentation.

If, **Switch**, and **Switch Case** blocks are now searched. See the “Check usage of Port and Subsystems” and “Check usage of Signal Routing blocks” checks for more information.

R2010bSP1

Version: 3.0.1

Bug Fixes

R2010b

Version: 3.0

New Features

Bug Fixes

Compatibility Considerations

New Functions for Extraction, Signal Logging, Harness Generation, and Test Execution for Model Components

Several new functions allow you to test the following model components:

- Models referenced from Model blocks
- Atomic subsystems
- Stateflow atomic subcharts

Using these functions, you can create and execute test cases that verify components in order to:

- Debug the component
- Test the robustness of the component
- Achieve 100% model coverage for the component

The new functions perform the following tasks.

Use...	To...
slvnvextract	Extract an atomic subsystem or Stateflow atomic subchart into a new model. This allows you to apply testing and verification techniques that use models or an isolated model component.
slvnvlogsignals	Simulate a Simulink model and log input signals to a model referenced by a Model block. This allows you to capture component inputs when you run a closed-loop system simulation.
slvnvmakeharness	Create a harness model for a component, using logged input signals if specified, or using the default signals. Harness models support graphical editing of input values using Signal Builder blocks.
slvnvmergeharness	Merge test cases and initializations from multiple harness models into a single harness model.
slvnvruntest	Simulate a model, executing the specified test cases to record model coverage and output values.
slvnvruncgvtest	Invoke the Code Generation Verification (CGV) API and execute the specified test cases on the generated code for the model.

New rmi Option to Validate Links in External Requirements Documents

The rmi function has a new option, `'checkdoc'`, that allows you to validate links from Microsoft Word or IBM Rational DOORS requirements documents to Simulink models.

`rmi` generates a report of the problems that it detects. The report includes hyperlinks that navigate to the requirements with problems so you can investigate and fix any invalid links.

For more information, see Checking and Fixing ActiveX-Based Links in Requirements Documents.

New Capability to Link a Requirement to Multiple Model Objects

The Requirements Management Interface (RMI) allows you to establish links to selected requirements in Microsoft Word, Microsoft Excel, or IBM Rational DOORS. When you create a link to a selected requirement for multiple model objects, the RMI inserts a single navigation control into the requirements document that links to all the selected model objects. When you activate the navigation control, all linked model objects are highlighted.

Viewing Requirements Linked to Model Objects

If your model, or blocks in your model, has links to requirements in external documents, you can now perform the following tasks without a Simulink Verification and Validation license:

- Highlight objects in a model that have links to requirements
- View information about a requirement
- Navigate from a model object to associated requirements
- Filter requirements highlighting based on keywords

Model Coverage Support for Multiple Instances of a Referenced Model

The Simulink Verification and Validation software records coverage for all instances of a referenced model in Normal simulation mode. The software combines the coverage data from each instance and reports one set of coverage data for the referenced model.

Model Coverage Support for Stateflow Atomic Subcharts

The Simulink Verification and Validation software records coverage for Stateflow atomic subcharts. Atomic subcharts allow you to reuse the same state or subchart across multiple Stateflow charts.

Checking Systems Programmatically With the Model Advisor

Previously, you could check systems programmatically using the `Simulink.ModelAdvisor` class, requiring many steps to check only one system at a time. Now you can check systems using the ModelAdvisor.run function. Using this function, you can create a MATLAB file for distribution that specifies a list of checks or a Model Advisor configuration file to run against any number of systems simultaneously.

Additionally, the following functions are available for reviewing results. You can use these functions to determine which checks failed on which systems and archive the results for accessing at a later time.

Use...	To...
ModelAdvisor.run	Run Model Advisor checks on systems.
ModelAdvisor.summaryReport	Open Model Advisor Command-Line Summary report in a web browser.
view	View Model Advisor check results for a system in a web browser.
viewReport	View Model Advisor run results for a system in a web browser, the Model Advisor GUI, or the Command Window.

If you have a Parallel Computing Toolbox license and a multicore machine, to improve performance, you can run the Model Advisor on multiple systems in parallel.

For details, see the ModelAdvisor.run reference page and Checking Systems Programmatically in the Simulink Verification and Validation documentation.

MathWorks Check IDs for the Model Advisor Updated

The check IDs for MathWorks checks in the Model Advisor are updated to use a consistent format. To easily determine check IDs, use the:

- Model Advisor context menu to return the check ID to the workspace.
- ModelAdvisor.lookupCheckID function to look up the new check ID.

For details, see Finding Check IDs in the Simulink Verification and Validation documentation.

Compatibility Considerations

The Model Advisor continues to recognize check IDs that were available prior to R2010b. Using the previous IDs in scripts and functions requires more processing time than if you use the new check IDs. To avoid a performance impact, update scripts and functions to use the new check IDs.

In prior releases, the following `Simulink.ModelAdvisor` methods returned the previous check IDs. In R2010b, the methods return the new check IDs:

```
getCheckAll
getCheckForGroup
getCheckForTask
getSelectedCheck
verifyCheckRan
```

New and Enhanced Model Advisor Checks

The Simulink Model Advisor includes new and updated verification and validation checks.

MathWorks Automotive Advisory Board Checks

The **MathWorks Automotive Advisory Board** subfolder now includes Check transition actions in Stateflow charts. Use this check to review your system against MAAB guideline db_0151: State machine patterns for transition actions.

For more information, see MathWorks Automotive Advisory Board Checks in the Simulink Verification and Validation User's Guide.

R2010a

Version: 2.7

New Features

Bug Fixes

Compatibility Considerations

Coverage Support for Dead Zone Block

The Simulink Verification and Validation software now records coverage for the Dead Zone block.

Coverage Support for the Simulink Design Verifier Functions in MATLAB Function Blocks

The Simulink Verification and Validation software now records coverage for the following Simulink Design Verifier functions in MATLAB Function blocks:

- sldv.assume
- sldv.condition
- sldv.prove
- sldv.test

Coverage Support for Trigger Ports in Model Blocks

The Simulink Verification and Validation software records coverage for a triggered Model block.

Requirements Support for MuPAD Documents

The Requirements Management Interface (RMI) now allows you to create requirements links from Simulink model objects to link targets in MuPAD® notebooks.

Improved Reports About Requirements Links in Models

In the requirements reports, the Requirements Management Interface (RMI) reporting capability includes the following improvements:

- Navigation to Simulink model objects with requirements
- Improved table design in requirements reports to accommodate additional detail about requirements

New Capability to Tag and Filter Model Requirements

The Requirements Management Interface (RMI) now allows filtered highlighting and reporting through the user tag attribute. Use this feature to review links with certain associated user tags. You can configure your settings to automatically apply specified tags when using selection-based linking. Manage these user tags with the new rmitag function.

Improved IBM Rational DOORS Synchronization

The Requirements Management Interface (RMI) synchronization of Simulink models with the IBM Rational DOORS software now provides control of change propagation for requirements links created in DOORS or Simulink. You can configure the synchronization to remove outdated links in DOORS whose corresponding links you have already deleted from Simulink. You can also configure the synchronization to remove outdated links in Simulink whose corresponding links you have already deleted from DOORS. When you delete objects with requirements from your Simulink model, you can automatically remove links in DOORS.

Function rminav Removed

The Simulink Verification and Validation software no longer supports the `rminav` function. To view model requirements, open the Model Explorer by selecting **View > Model Explorer** or entering the `daexplr` command. In the Model Explorer, click the **Highlight items with requirements on model** icon.

Compatibility Considerations

In R2009b, the `rminav` command opened the Model Explorer.

New and Enhanced Model Advisor Checks

The Simulink Model Advisor includes new and updated verification and validation checks.

DO-178B Checks

The **DO-178B Check** subfolder now includes:

- Check state machine type of Stateflow charts
- Check Stateflow charts for ordering of states and transitions
- Check Stateflow debugging settings

The checks in the **DO-178B Checks** subfolder are reformatted for consistency in the results format.

For more information, see DO-178B Checks in the Simulink Verification and Validation documentation.

IEC 61508 Checks

The **IEC 61508 Checks** subfolder now includes Check state machine type of Stateflow charts.

For more information, see IEC 61508 and ISO 26262 Checks in the Simulink Verification and Validation documentation.

MathWorks Automotive Advisory Board Checks

The **MathWorks Automotive Advisory Board** subfolder now includes:

- Check usage of return values from a graphical function in Stateflow charts
- Check for pointers in Stateflow charts
- Check for event broadcasts in Stateflow charts
- Check for MATLAB expressions in Stateflow charts
- Check for bitwise operations in Stateflow charts
- Check for comparison operations in Stateflow charts
- Check for unary minus operations on unsigned integers in Stateflow charts
- Check for equality operations between floating-point expressions in Stateflow charts

The checks in the **MathWorks Automotive Advisory Board** subfolder are reformatted for consistency in the results format.

For more information, see MathWorks Automotive Advisory Board Checks.

Modeling Guidelines Documentation Moved into Simulink Documentation

The *Modeling Guidelines for High-Integrity Systems* and *MathWorks Automotive Advisory Board Control Algorithm Modeling Guidelines Using MATLAB, Simulink, and Stateflow* documentation is now in the Simulink documentation.

For more information, see Modeling Guidelines in the Simulink documentation.

R2009bSP1

Version: 2.6.1

Bug Fixes

R2009b

Version: 2.6

New Features

Bug Fixes

Compatibility Considerations

New Coverage Reporting for Variable-Size Signals

In R2009b, signal sizes in Simulink models can change during execution to facilitate modeling systems with varying environments, resources, and constraints. The Simulink Verification and Validation software can report the minimum, maximum, and allocated space for all variable-size signals in a model, as recorded during simulation.

To enable this feature, in the Coverage Settings dialog box, on the **Coverage** tab, select the **Signal Size** option.

Model Coverage for Simulink Design Verifier Blocks

The Simulink Verification and Validation model coverage tool can collect model coverage data for Simulink Design Verifier blocks. Using this feature during simulation, you can verify Simulink Design Verifier generated test harnesses and user-defined requirements. The model coverage tool collects model coverage data for the following Simulink Design Verifier blocks:

- Proof Assumption
- Proof Objective
- Test Condition
- Test Objective

To enable this feature, in the Coverage Settings dialog box, on the **Coverage** tab, select **Simulink Design Verifier**.

Force Block Reduction Off During Simulation

The Simulink Verification and Validation software can override the value of the Simulink **Block reduction** parameter. When you set this option, the simulation records coverage for all blocks in the model.

To enable this feature, in the Coverage Settings dialog box, on the **Options** tab, select **Force block reduction off**.

Compatibility Considerations

Previously, model coverage ignored the value of the Simulink **Block reduction** parameter, creating incompatibilities between the model coverage data and the generated code coverage data. In R2009b, you can select **Force block reduction off** to override the **Block reduction** parameter and collect coverage for all blocks in the model.

Improved RMI Highlighting and Reporting

The Requirements Management Interface (RMI) has the following improvements to its highlighting and reporting capabilities:

- Differential highlighting of blocks and subsystems to easily locate model objects with requirements links
- Real-time update of highlighted view to reflect the current state of the model requirements links
- Improved requirements report that includes a document summary, and details about signal group requirements and objects with missing requirements
- New interface for modifying settings for the requirements report generation that is available to users who do not have a Simulink Report Generator license
- IBM Rational DOORS module paths stored with the requirements information for easier review. For requirement links created before R2009b, the module path information is still saved only in the DOORS module.

Create Custom Configurations Using the Model Advisor Configuration Editor

The Model Advisor Configuration Editor expedites creating and deploying custom configurations of the Model Advisor. The Configuration Editor assists with:

- Adding MathWorks and custom checks to your custom configuration with drag and drop.
- Creating new folders in your custom configuration from a toolbar icon.
- Deploying custom configurations in a MAT file.

For more information, see Organizing Checks and Folders Using the Model Advisor Configuration Editor and How to Deploy Custom Configurations in the Simulink Verification and Validation documentation.

Licensing Revised for the Model Advisor

Previously, the Model Advisor populated the list of checks and folders based on the licenses available to you. When you ran checks, the software did not require a license checkout. In R2009b, the Model Advisor performs a checkout of the Simulink Verification and Validation license when you run Simulink Verification and Validation checks. For a list of Simulink Verification and Validation checks, see Model Advisor Checks in the Simulink Verification and Validation documentation.

Compatibility Considerations

The Model Advisor performs a checkout of a Simulink Verification and Validation license when you run:

- Checks in the Simulink Verification and Validation folder
- Custom checks

If you do not have a Simulink Verification and Validation license available, when you run the check, you see an error message that the required license is not available.

New and Enhanced Model Advisor Checks

The Simulink Model Advisor includes new and updated verification and validation checks.

DO-178B Checks

The **DO-178B Check** subfolder now includes Check safety-related diagnostic settings for model initialization.

The checks in the **DO-178B Checks** subfolder are reformatted for a consistent look and feel to the results.

For more information, see DO-178B Checks in the Simulink Verification and Validation User's Guide.

IEC 61508 Checks

The **IEC 61508 Checks** subfolder now includes Check for model objects that do not link to requirements.

For more information, see IEC 61508 and ISO 26262 Checks in the Simulink Verification and Validation documentation.

Modeling Guidelines for High-Integrity Systems

MathWorks intends this document for engineers developing models and generating code for high-integrity systems using Model-Based Design with MathWorks products. This document describes creating Simulink models that are complete, unambiguous, statistically deterministic, robust, and verifiable. The document focus is on model settings, block usage, and block parameters that impact simulation behavior or code generated by the Real-Time Workshop® Embedded Coder product.

These guidelines do not assume that you use a particular safety or certification standard. The guidelines reference some safety standards where applicable, including DO-178B, IEC 61508, and MISRA C.

You can use the Model Advisor to support adhering to these guidelines. Each guideline lists the checks that are applicable to that guideline.

For more information, see “Modeling Guidelines for High-Integrity Systems” in the Simulink Verification and Validation documentation.

R2009a

Version: 2.5

New Features

Bug Fixes

Recording Coverage for External MATLAB Files

MATLAB Function blocks and functions can call external program files. The Model Coverage Tool extends the model coverage capabilities to record coverage for these files. By default, this option is disabled; coverage is not collected for external program files.

Enable this option if you want the software to collect and report coverage for all program files associated with the MATLAB Function blocks. To enable this option, in the Coverage Settings dialog box, select **Coverage for MATLAB files**.

API for Customizing Model Advisor Results

In R2009a, there is a new API to customize the results of the Model Advisor for a common look and feel for all of your customized checks. The API includes templates for you to organize the results in tables or lists. For more information, see Customizing the Model Advisor in the Simulink Verification and Validation User's Guide.

New and Enhanced Model Advisor Checks

The Simulink Model Advisor includes new and updated verification and validation checks.

DO-178B Checks

The **DO-178B Check** subcategory includes the following new checks:

- Check for blocks that do not link to requirements
- Check usage of Math blocks

For more information, see DO-178B Checks in the Simulink Verification and Validation documentation.

IEC 61508 Checks

The checks in the **IEC 61508 Checks** subfolder are reformatted for a consistent look and feel to the results. For more information, see IEC 61508 and ISO 26262 Checks in the Simulink Verification and Validation documentation.

Modeling Standards for MAAB Checks

There is a new **Modeling Standards for MAAB** subfolder in the **By Task** folder of the Model Advisor, which includes subfolders that organize the MAAB checks so you can more easily determine which checks you want to run.

MathWorks Automotive Advisory Board Control Algorithm Modeling Guidelines Using MATLAB, Simulink, and Stateflow Included in Help

The MathWorks Automotive Advisory Board (MAAB) involves major automotive original equipment manufacturers (OEMs) and suppliers in the process of evolving MathWorks controls, simulation, and code generation products, including the Simulink, Stateflow, and Real-Time Workshop products. An important result of the MAAB has been the "MathWorks Automotive Advisory Board Control Algorithm Modeling Guidelines Using MATLAB, Simulink, and Stateflow." Help for the Simulink Verification and Validation product now includes these guidelines. The MAAB guidelines link to relevant Model Advisor MAAB check help and MAAB check help links to relevant MAAB guidelines.

For more information on the MAAB guidelines, see "MathWorks Automotive Advisory Board Control Algorithm Modeling Guidelines Using MATLAB, Simulink, and Stateflow". For information on the Model Advisor MAAB checks, see "MathWorks Automotive Advisory Board Checks."

R2008b

Version: 2.4

New and Enhanced Model Advisor Checks

Context-sensitive help is now available for all Model Advisor checks.

DO-178B Checks

The **DO-178B Check** subcategory includes these new and enhanced checks:

- Check safety-related diagnostic settings for saving — This new check helps enforce and document adherence to coding standards for DO-178B, Software Considerations in Airborne Systems and Equipment Certification.
- Check safety-related diagnostic settings for signal data — This check now checks whether the parameter `SignalRangeChecking` is set to `error`.
- New Action box for the following checks that you can use to automatically fix failures:
  - Check safety-related optimization settings
  - Check safety-related diagnostic settings for solvers
  - Check safety-related code generation settings
  - Check safety-related diagnostic settings for compatibility
  - Check safety-related diagnostic settings for bus connectivity
  - Check safety-related diagnostic settings for signal connectivity
  - Check safety-related diagnostic settings for data store memory
  - Check safety-related diagnostic settings for data used for debugging
  - Check safety-related diagnostic settings for parameters
  - Check safety-related diagnostic settings for signal data
  - Check safety-related diagnostic settings that apply to function-call connectivity
  - Check safety-related diagnostic settings for model referencing
  - Check safety-related model referencing settings
  - Check safety-related diagnostic settings for sample time
  - Check safety-related diagnostic settings for type conversions
  - Check safety-related diagnostic settings for saving

For more information, see DO-178B Checks in the Simulink Verification and Validation documentation.

IEC 61508 Checks

The **IEC 61508 Checks** subcategory includes a new check, Check usage of Stateflow, that helps enforce and document compliance with coding standards for functional safety in electrical, electronic, and programmable electronic systems, as the IEC 61508 standard outlines.

For more information, see IEC 61508 and ISO 26262 Checks in the Simulink Verification and Validation documentation.

MathWorks Automotive Advisory Board Checks

The **MathWorks Automotive Advisory Board Checks** subcategory includes new checks that have been implemented to support the MathWorks Automotive Advisory Board (MAAB) 2.0 rules:

- Check for difference in font and font sizes
- Check Transition orientations in Flowcharts
- Check for display of non-default block attributes
- Check signal line labels
- Check for propagated labels on Signal lines
- Check default Transition placement on Stateflow Charts
- Check setting Stateflow Graphical Function return value
- Check for proper use of Switch Blocks
- Check for proper use of Signal Busses and Mux block usage

For more information, see MathWorks Automotive Advisory Board Checks in the Simulink Verification and Validation documentation.

Improved Selection-Based Linking

To create links between a model and a Microsoft Word or Microsoft Excel file, you can open the file from within both the model and the requirements document. The Requirements Management Interface (RMI) establishes communication with the application when you initiate selection-based linking.

For more information, see Links Between Models and Requirements in the Simulink Verification and Validation documentation.

Enhanced Model Coverage Report

The Model Coverage report now includes information about:

- Model being analyzed
- Simulation optimization options
- Model coverage options

For more information, see Creating Requirements Reports in the Simulink Verification and Validation documentation.

R2008a+

Version: 2.3.1

Bug Fixes

R2008a

Version: 2.3

New Features

Bug Fixes

Enhanced Coverage Support for Referenced Models

The previous release introduced coverage support for referenced models, but only by means of functions that you invoke programmatically. This release includes the following enhancements to coverage support for referenced models:

- Redesigned Coverage Settings dialog box now includes controls that allow you to enable coverage for referenced models graphically.
- New Coverage by Model summary that allows you to navigate easily to coverage reports associated with referenced models in a hierarchy.

New Requirements Submenu in Model Explorer

Full access to the Requirements Management Interface is now available from within the Model Explorer. In this release, a **Requirements** submenu replaces the **Edit requirements** item that appears in Model Explorer context menus. This new submenu provides several options for adding requirements to model elements and specifying link preferences. This same submenu is available by right-clicking objects in a Simulink model or a Stateflow chart.

Model Advisor Checks for Modeling Standards Enhanced

The Simulink Model Advisor has been enhanced with new and updated verification and validation checks.

- In the Task Hierarchy tree, the check category **By Product** > **Simulink Verification and Validation** > **Modeling Standards** includes the new subcategory **IEC 61508 Checks**.
- The **IEC 61508 Checks** subcategory includes new checks that help enforce and document adherence to coding standards for functional safety in electrical, electronic, and programmable electronic systems, as the IEC 61508 standard outlines. For descriptions of the IEC 61508 checks, see Model Advisor Checks in the Simulink Verification and Validation documentation.
- The IEC 61508 checks also appear under the new category **By Task** > **Model-Based Design for IEC 61508**.
- Some **MathWorks Automotive Advisory Board Checks** have been updated.

For Information on How To...	See...
Develop models and code that comply with industry standards and guidelines, such as the IEC 61508 standard and MathWorks Automotive Advisory Board (MAAB) guidelines	Guidelines and Standards in the Real-Time Workshop Embedded Coder documentation
Use the Model Advisor	Consulting the Model Advisor in Simulink User's Guide
Customize the Model Advisor with your own tasks and checks	Customizing the Model Advisor in the Simulink Verification and Validation User's Guide

API for Customizing Model Advisor Enhanced

R2008a provides an enhanced API that allows you to customize the behavior of the Model Advisor for checking and reporting on how well your Simulink models are optimized for simulation and code generation. The API provides classes and functions for defining custom checks and tasks, and writing your own callback functions. For more information, see Customizing the Model Advisor in the Simulink Verification and Validation User's Guide.

R2007b+

Version: 2.2.1

Bug Fixes

R2007b

Version: 2.2

New Features

Bug Fixes

Model Coverage Support for Referenced Models

This release introduces model coverage support for referenced models that operate in Normal mode. That is, you can now record coverage for Model blocks whose **Simulation mode** parameter specifies `Normal`. Toward that end, this release provides three new functions:

- cvsimref
- cv.cvdatagroup
- cv.cvtestgroup

For more information, see Using Model Coverage Commands for Referenced Models in the Simulink Verification and Validation User's Guide.

Model Advisor Checks for Modeling Standards Enhanced

The Simulink Model Advisor has been enhanced with new and updated verification and validation checks.

- In the Task Hierarchy tree, the check category **By Product** > **Simulink Verification and Validation** > **Modeling Standards** includes two new subcategories: **DO-178B Checks** and **MathWorks Automotive Advisory Board Checks**.
- The **DO-178B Checks** subcategory includes new checks that help enforce and document adherence to coding standards for DO-178B, Software Considerations in Airborne Systems and Equipment Certification. If a piece of software will be flying on a commercial aircraft, the development and verification process must comply with DO-178B. For descriptions of the DO-178B checks, see Model Advisor Checks in the Simulink Verification and Validation documentation.
- The DO-178B checks also appear under the new category **By Task** > **Safety Critical for DO-178B**.
- Checks listed under **By Product** > **Simulink Verification and Validation** > **Modeling Standards** in earlier releases, appear under the new subcategory **MathWorks Automotive Advisory Board Checks**. These checks verify that Simulink block diagrams adhere to style guidelines published by the MathWorks Automotive Advisory Board (MAAB).
- Some **MathWorks Automotive Advisory Board Checks** have been updated. For example, the check for “hd_0001, Prohibited Simulink Sink and Data Store blocks”

was modified to allow Data Store Memory, Data Store Read, and Data Store Write blocks and the check for “db_0146, Triggered or enabled subsystems” was modified to include For Iterator, While Iterator, and Case blocks.

For Information on How To...	See...
Use the Model Advisor	Consulting the Model Advisor in Simulink User's Guide
Customize the Model Advisor with your own tasks and checks	Customizing the Model Advisor in the Simulink Verification and Validation User's Guide

API for Customizing Model Advisor HTML Reports

This release provides a new API that allows you to customize Model Advisor HTML reports. The API provides classes and methods for formatting custom HTML reports, making it easier to create, understand, and maintain Model Advisor HTML reports. See Formatting Model Advisor Results in the Simulink Verification and Validation User's Guide.

R2007a+

Version: 2.1.1

Bug Fixes

R2007a

Version: 2.1

Bug Fixes

R2006b

Requirement Links Checking

This release introduces Model Advisor tasks for ensuring the consistency and validity of requirement links in a Simulink block diagram. These checks enable you to identify and fix the following situations:

- The requirements document is missing.
- The location (e.g., bookmark, line number, anchor) specified within the requirements document is invalid.
- The description of a selection-based link does not match its requirements document text.
- The requirements document path does not match your preferred path type (i.e., absolute or relative).

See Consulting the Model Advisor in Simulink User's Guide for more information about using Model Advisor.

Modeling Standards Checking

This release introduces Model Advisor tasks for ensuring that a Simulink block diagram adheres to the style guidelines published by the MathWorks Automotive Advisory Board (MAAB). See Consulting the Model Advisor in Simulink User's Guide for more information about using Model Advisor. For information about the MAAB modeling style guidelines, see `http://www.mathworks.com/industries/auto/maab.html`.

R2006a

Version: 1.1.2

No New Features or Changes