

# Reactis V2013

Released August 14, 2013



# New Simulink Support

- ▶ Initial support for R2013a
- ▶ Support for .slx Format
  - ▶ Introduced in R2012a to replace .mdl format
  - ▶ XML-based format
- ▶ Support Simulink *configuration set reference* concept

# Multiple Condition Coverage (MCC)

MCC metric tracks if all combinations of condition outcomes for a decision have been exercised.

# Multiple Condition Coverage (MCC)

MCC metric tracks if all combinations of condition outcomes for a decision have been exercised. For decision  $A \ \&\& \ B \ \&\& \ C$ , MCC targets are:

# Multiple Condition Coverage (MCC)

MCC metric tracks if all combinations of condition outcomes for a decision have been exercised. For decision  $A \ \&\& \ B \ \&\& \ C$ , MCC targets are:

## No Short-Circuiting

A	B	C	Decision
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	F
T	F	F	F
T	F	T	F
T	T	F	F
T	T	T	T

# Multiple Condition Coverage (MCC)

MCC metric tracks if all combinations of condition outcomes for a decision have been exercised. For decision  $A \ \&\& \ B \ \&\& \ C$ , MCC targets are:

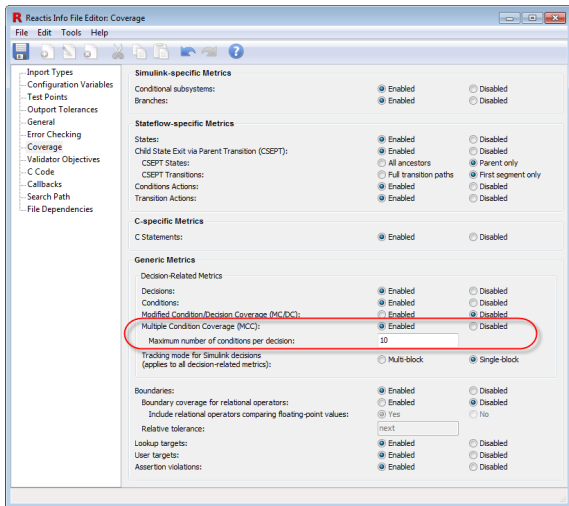
## No Short-Circuiting

A	B	C	Decision
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	F
T	F	F	F
T	F	T	F
T	T	F	F
T	T	T	T

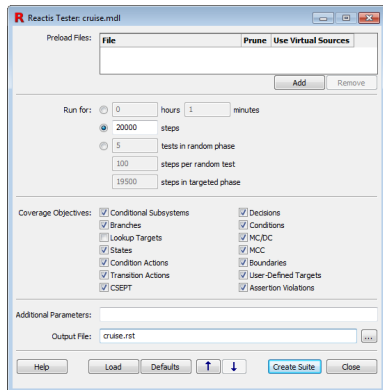
## Short-Circuiting

A	B	C	Decision
F	x	x	F
T	F	x	F
T	T	F	F
T	T	T	T

# MCC Now Supported by Reactis

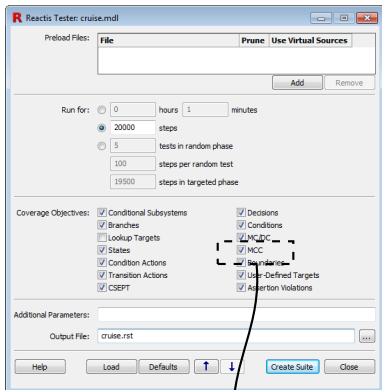


# MCC Support in Reactis Tester



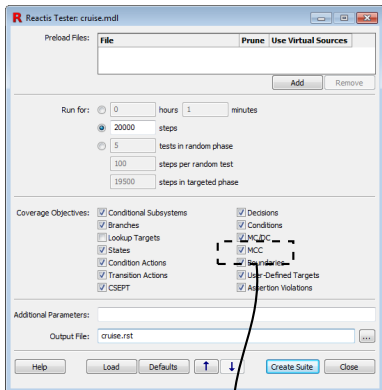


# MCC Support in Reactis Tester

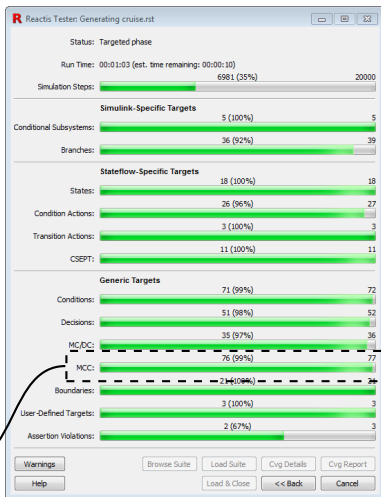


Tester attempts to cover MCC targets

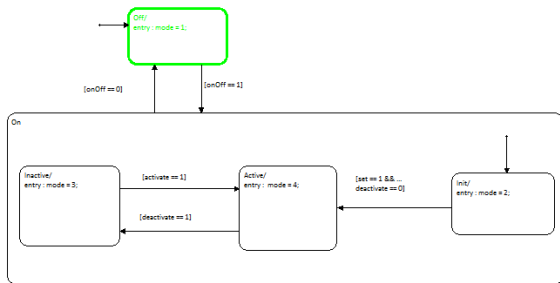
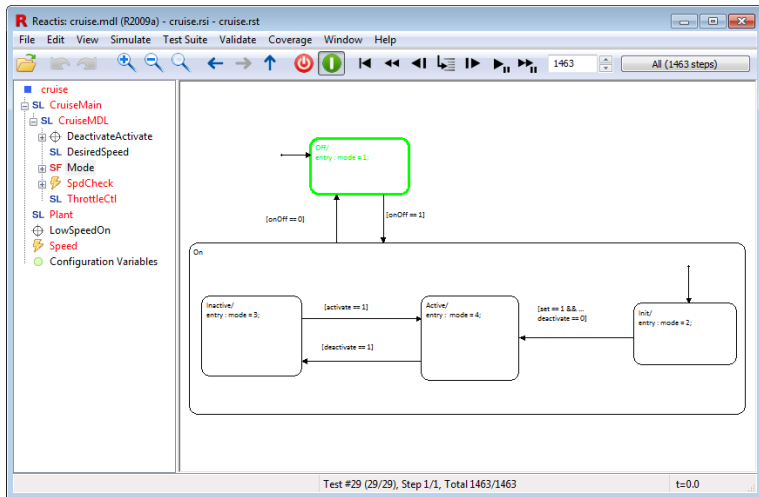
# MCC Support in Reactis Tester



Tester attempts to cover MCC targets



# Tracking MCC in Simulator



Test #29 (29/29), Step 1/1, Total 1463/1463

t=0.0

# Tracking MCC in Simulator

The screenshot displays the Reactis simulator window for a file named 'cruise.mdl (R2009a)'. The interface includes a menu bar (File, Edit, View, Simulate, Test Suite, Validate, Coverage, Window, Help), a toolbar with navigation and simulation controls, and a status bar at the bottom showing 'Test #29 (29/29), Step 1/1, Total 1463/1463' and 't=0.0'.

On the left, a project tree lists the following components:

- cruise
  - SL CruiseMain
    - SL CruiseMDL
      - DeactivateActivate
        - SL DesiredSpeed
        - SF Mode
        - SpdCheck
        - SL ThrottleCtl
      - SL Plant
        - LowSpeedOn
        - Speed
        - Configuration Variables

The main workspace shows a state machine diagram. A state box at the top is labeled 'Off/ entry : mode = 1;'. It has a transition to a state box labeled 'Decision with MCC targets' with the guard '[onOff == 0]'. From the 'Decision with MCC targets' state, there are two outgoing transitions: one to the 'Off/ entry : mode = 1;' state with guard '[onOff == 1]', and another to a state box labeled 'init/ entry : mode = 2;'. From the 'init/ entry : mode = 2;' state, there is a transition back to the 'Decision with MCC targets' state with guard '[set == 1 && ... deactivate == 0]'. The 'Decision with MCC targets' state also has a self-loop transition labeled 'inactive/ entry : mod...'. The state 'Decision with MCC targets' is highlighted with a blue border.

# Tracking MCC in Simulator

The screenshot displays the Reactis simulator window for a file named 'cruise.mdl (R2009a)'. The interface includes a menu bar (File, Edit, View, Simulate, Test Suite, Validate, Coverage, Window, Help), a toolbar with navigation and simulation controls, and a progress indicator showing '1463' steps out of 'All (1463 steps)'. On the left, a project tree lists components: 'cruise', 'SL CruiseMain', 'SL CruiseMDL', 'DeactivateActivate', 'SL DesiredSpeed', 'SF Mode', 'SpdCheck', 'SL ThrottleCtl', 'SL Plant', 'LowSpeedOn', 'Speed', and 'Configuration Variables'. The main workspace shows a state machine diagram with a central state box labeled 'Decision with MCC targets'. This state is reached from an 'On' state and has transitions to 'mode = 1' and 'mode = 2'. A 'Toggle Breakpoint' button is visible, with a 'View Coverage Details' button below it. The status bar at the bottom indicates 'Test #29 (29/29), Step 1/1, Total 1463/1463' and 't=0.0'.

# Tracking MCC in Simulator

The screenshot displays the Reactis simulator interface. The main window shows a state machine diagram with a state labeled "Decision with MCC targets". A green box highlights the state's entry condition: "On/entry : mode = 1;". A transition labeled "[onOff == 1]" leads from this state to another state with the entry condition "init/entry : mode = 2;". A "Toggle Breakpoint" button is visible, with a "View Coverage Details" button below it. The status bar at the bottom indicates "Test #29 (29/29), Step 1/1, Total 1463/1463" and "t=0.0".

The "Coverage Details" window is open, showing a table with the following data:

Decision		Condition	Condition		MC/DC		MC/DC	
True	False		True	False	True	False	True	False
1/30	1/4	set==1.0 deactivate==0.0	1/20	1/4	TT: 1/30	Fx: 1/4	TT: 1/30	TF: 1/20

# Tracking MCC in Simulator

The image displays a simulator interface with two 'Coverage Details' windows and a state transition diagram. The diagram includes a state labeled 'Decision with MCC targets' and a 'Toggle Breakpoint / View Coverage Details' button. A green box highlights a state transition labeled 'OH/ entry : mode = 1;'. The status bar at the bottom shows 'Test #29 (29/29), Step 1/1, Total 1463/1463' and 't=0.0'.

**Left Coverage Details Window:**

Decision		Condition	Condition	Condition	MC/DC	MC/DC
True	False		True	False	True	False
1/30	1/4	set==1.0	1/20	1/4	TT: 1/30	Fx: 1/4
		deactivate==0.0	1/30	1/20	TT: 1/30	TF: 1/20

**Right Coverage Details Window:**

Click on column headers to enable filters Clear Filter

set==1.0	deactivate==0.0	Decision	Covered
False	x	False	1/4
True	False	False	1/20
True	True	True	1/30

**State Transition Diagram:**

- State: **Decision with MCC targets**
- Transitions:
  - From 'inactive/ entry : mod...' to 'Decision with MCC targets' (labeled 'mode = 1;')
  - From 'Decision with MCC targets' to 'Toggle Breakpoint / View Coverage Details' (labeled 'mode = 1;')
  - From 'Toggle Breakpoint / View Coverage Details' to 'Decision with MCC targets' (labeled 'mode = 1;')
  - From 'Decision with MCC targets' to 'init/ entry : mode = 2;' (labeled 'mode = 1;')
  - From 'init/ entry : mode = 2;' to 'Decision with MCC targets' (labeled 'mode = 2;')
- Condition: [set == 1.0 ... deactivate == 0]

# Tracking MCC in Simulator

The image displays a simulator interface with two 'Coverage Details' windows and a 'Decision with MCC targets' block. The top window shows a table of MCC targets, and the bottom window shows a table of coverage data for a specific target. The 'Decision with MCC targets' block is a state machine diagram with a 'Toggle Breakpoint' and 'View Coverage Details' button.

**Row for each MCC target**

Decision	MCC
True	False
1/30	1/4
	set==1.0
	deactivate==0.0
Condition True	Condition False
1/20	1/4
1/30	1/20
MC/DC True	MC/DC False
TT: 1/30	Fx: 1/4
TT: 1/30	TF: 1/20

**Decision with MCC targets**

set==1.0	deactivate==0.0	Decision	Covered
False	x	False	1/4
True	False	False	1/20
True	True	True	1/30

OH/ entry : mode = 1;

Toggle Breakpoint  
View Coverage Details

Test #29 (29/29), Step 1/1, Total 1463/1463 t=0.0



# MCC Filtering

Decision MCC

Click on column headers to enable filters Clear Filter

set==1.0	deactivate==0.0	Decision	Covered
False	x	False	1/4
True	False	False	1/20
True	True	True	1/30

Help Close

# MCC Filtering

*Click column header*

The image shows two side-by-side screenshots of a software window titled "Coverage Details". The window has a tab labeled "Decision" and "MCC". Below the tab, there is a text prompt "Click on column headers to enable filters" and a "Clear Filter" button. The main content is a table with columns: "set==1.0", "deactivate==0.0", "Decision", and "Covered".

In the left screenshot, the table contains three rows of data:

set==1.0	deactivate==0.0	Decision	Covered
False	x	False	1/4
True	False	False	1/20
True	True	True	1/30

In the right screenshot, the "Decision" column header is highlighted, and the table is filtered to show only rows where "Decision" is "True". The table now contains one row:

set==1.0	deactivate==0.0	T: Decision	Covered
True	True	True	1/30

A yellow callout box with blue text says "Only display rows with *True* in column". An arrow points from this box to the "T: Decision" header in the right screenshot. Another arrow points from the "Decision" header in the left screenshot to the callout box. The window also has "Help" and "Close" buttons at the bottom.

# MCC Filtering

*Click column header*

Click on column headers to enable filters

set==1.0	deactivate==0.0	Decision	Covered
False	x	False	1/4
True	False	False	1/20
True	True	True	1/30

Only display rows with *True* in column

Click column header again

Click on column headers to enable filters

set==1.0	deactivate==0.0	T: Decision	Covered
True	True	True	1/30

Close

Click on column headers to enable filters

set==1.0	deactivate==0.0	F: Decision	Covered
False	x	False	1/4
True	False	False	1/20

Only display rows with *False* in column

# MCC Considerations

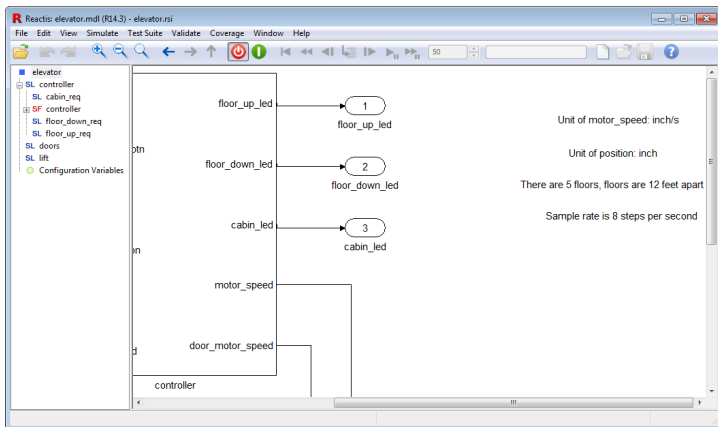
- ▶ Decision with  $n$  conditions has  $2^n$  MCC targets.
- ▶ When short-circuiting is enabled, many fewer MCC targets. Number is between MC/DC ( $n+1$ ) and  $2^n$ .
- ▶ Even if 100% MCC is not goal, MCC coverage details can be helpful in obtaining MC/DC.
- ▶ In model-specific settings (*Edit* → *Coverage...*), can set upper-bound on number of conditions in a decision; if exceeded MCC will not be tracked.

# Text Search Enhancement

Text search now searches Simulink annotations

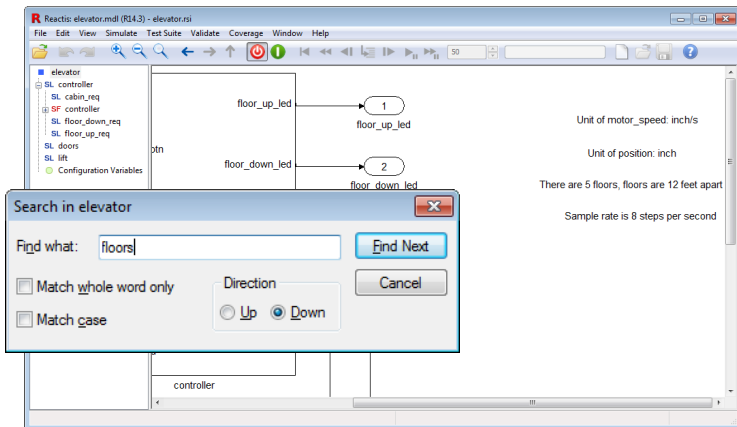
# Text Search Enhancement

Text search now searches Simulink annotations



# Text Search Enhancement

Text search now searches Simulink annotations



# Text Search Enhancement

Text search now searches Simulink annotations

The screenshot displays the Simulink environment for a model named 'elevator.mdl'. A search dialog box titled 'Search in elevator' is open, showing the search term 'floors'. The dialog includes options for 'Match whole word only', 'Match case', and 'Direction' (set to 'Down'). The search results are displayed in the right-hand pane, showing the following text:

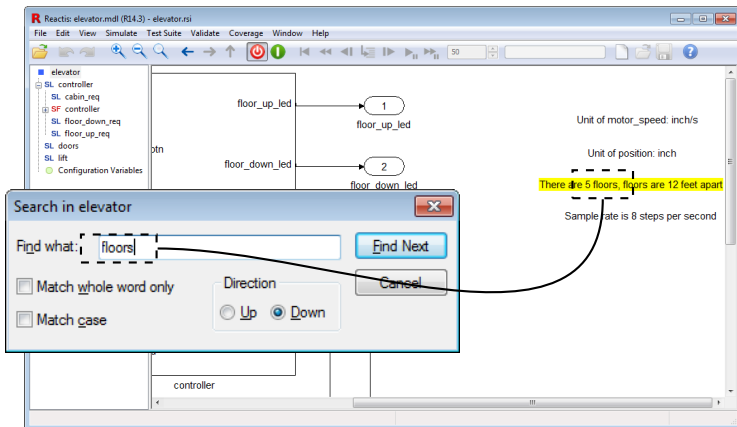
- Unit of motor\_speed: inch/s
- Unit of position: inch
- There are 5 floors, floors are 12 feet apart** (highlighted in yellow)
- Sample rate is 8 steps per second

The background Simulink diagram shows a block labeled 'controller' with two outputs: 'floor\_up\_led' and 'floor\_down\_led'. Each output is connected to a scope block labeled '1' and '2' respectively. The left-hand pane shows the model hierarchy, including 'SL controller', 'SL cabin\_req', 'SF controller', 'SL floor\_down\_req', 'SL floor\_up\_req', 'SL doors', and 'SL lift'.

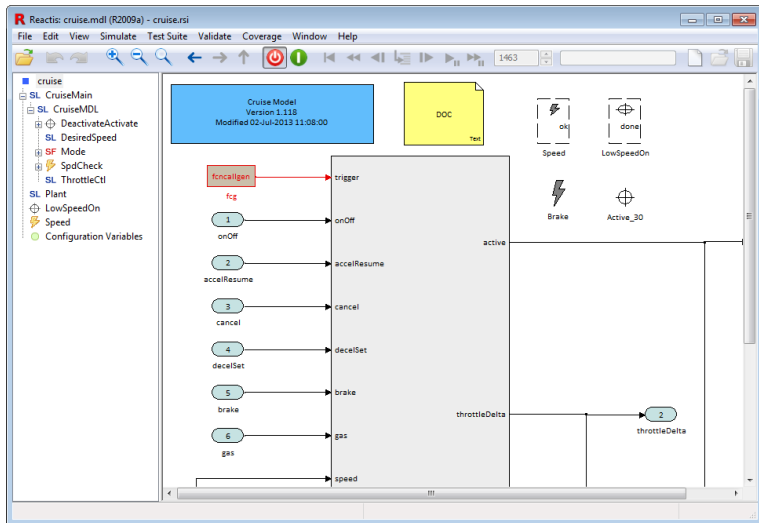


# Text Search Enhancement

Text search now searches Simulink annotations



# Display Simulink Block Colors



# Display Model Info and Doc Block Content

## Display contents of Model Info and Doc blocks

The screenshot displays the Reactis software interface for a model named 'cruise.mdl (R2009a) - cruise.rsi'. The main window shows a block diagram of the 'Cruise Model' with version 1.113, last modified on 10-Apr-2013 at 15:34:52. The diagram includes a 'trigger' block connected to an 'fncallgen' block, and a 'DOC' block. A 'DOC block' dialog box is open, displaying the following text:

**DOC block**

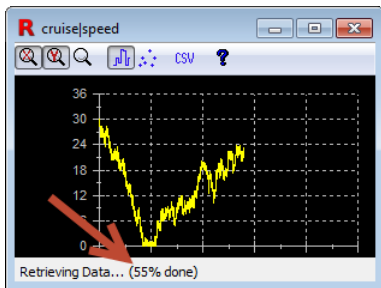
This model implements a cruise control to demonstrate the different capabilities offered by Reactis to test and validate models. Chapter 3 of the Reactis User's Guide describes how to use Reactis with this model.

The dialog box also features a 'Close' button.

The background interface includes a menu bar (File, Edit, View, Simulate, Test Suite, Validate, Coverage, Window, Help), a toolbar with various icons, and a left-hand pane showing a tree view of the model structure:

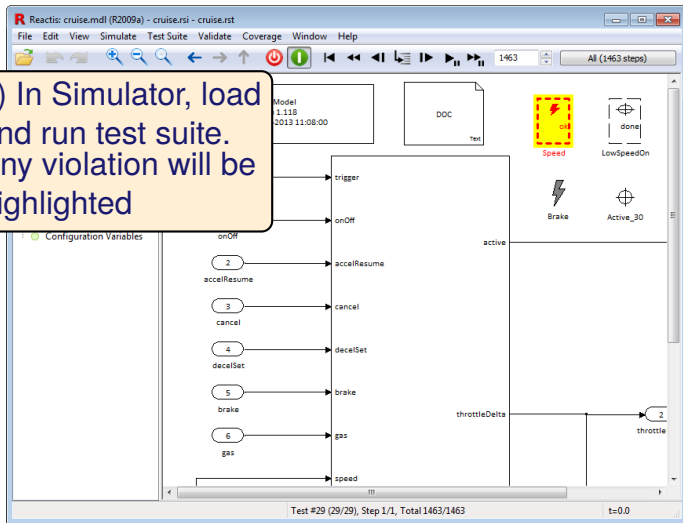
- cruise
  - SL CruiseMain
  - SL Plant
  - LowSpeedOn
  - Speed
  - Configuration Variables

# Show Progress when Opening Scope



# Run-to-Violation

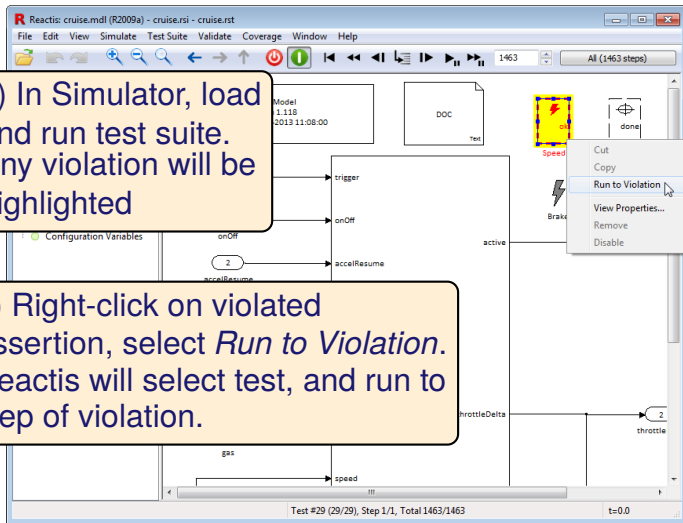
1) In Simulator, load and run test suite. Any violation will be highlighted



# Run-to-Violation

1) In Simulator, load and run test suite. Any violation will be highlighted

2) Right-click on violated assertion, select *Run to Violation*. Reactis will select test, and run to step of violation.



# Lookup Table Enhancements

- ▶ Enhancements to N-d lookup tables, Pre Lookup, Interpolation
  - ▶ Track coverage
  - ▶ Support more type combinations
- ▶ In coverage reporting, show breakpoint values

	$[-\text{inf}, c1]$	$[c1, c2]$	$[c2, c3]$	$[c3, +\text{inf}]$
$[-\text{inf}, r1]$	1/4	2/3	5/1	1/6
$[r1, r2]$	1/5	2/2	3/11	2/1
$[r2, +\text{inf}]$	1/1	1/3	1/2	1/7

Old

	$u2 < 1.0$	$u2 \geq 1.0$	$u2 \geq 10.0$	$u2 \geq 100.0$
$u1 < 1.0$	1/4	2/3	5/1	1/6
$u1 \geq 1.0$	1/5	2/2	3/11	2/1
$u1 \geq 10.0$	1/1	1/3	1/2	1/7

New

- ▶ Tester gets better coverage for lookup tables

# API Functions to Export Coverage Reports

- ▶ *rsTesterWithReport* produces a coverage report after a Tester run
- ▶ *rsSimRunSuiteWithReport* produces a coverage report after running a test suite in Simulator