

Reactis V2015

Released June 30, 2015



New Simulink Support

- ▶ R2015a.
- ▶ Simulink[®] Function block and Function Caller block and calling Simulink Function blocks from Stateflow[®].
- ▶ Extended subset of Embedded MATLAB[®] (EML)
 - ▶ Logical functions: xor, and, or, not, any, all.
 - ▶ Math functions: mod, rem.
 - ▶ logical function now accepts matrix arguments
 - ▶ Other functions: isnan, isinf, isfinite, return

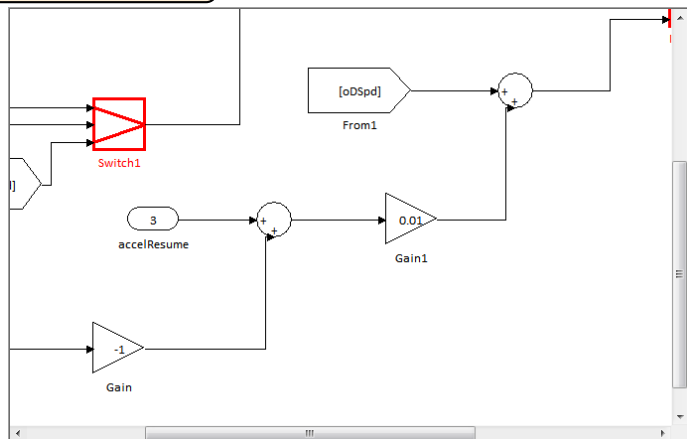
Excluding Coverage Targets

Any target can be excluded from coverage tracking.

- ▶ Tester will not attempt to exercise the target with generated tests.
- ▶ Simulator will not include the target in coverage reporting (including exported reports).

Excluding a Target

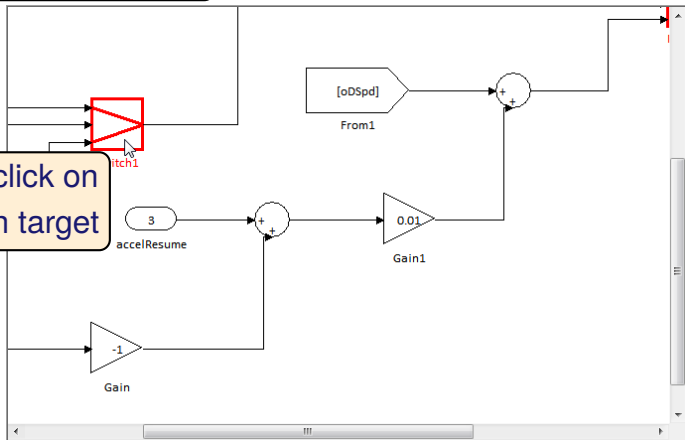
1) Enable Simulator



Excluding a Target

1) Enable Simulator

2) Right-click on block with target

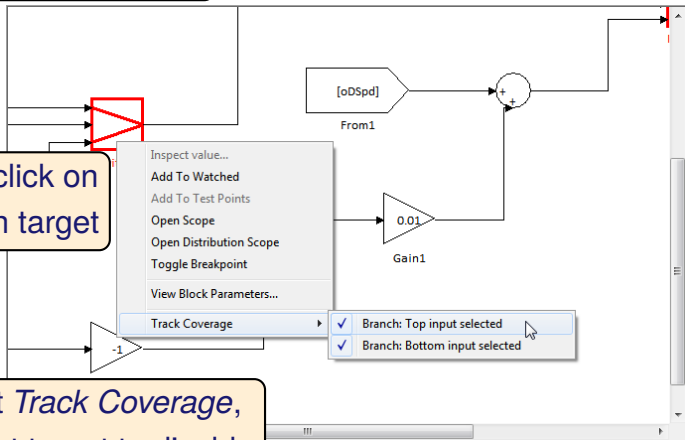


Excluding a Target

1) Enable Simulator

2) Right-click on block with target

3) Select *Track Coverage*, then select target to disable



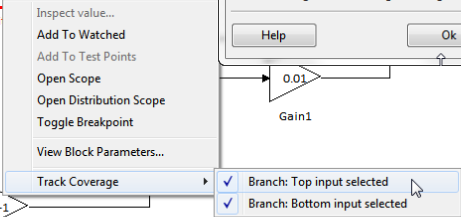
Excluding a Target

1) Enable Simulator

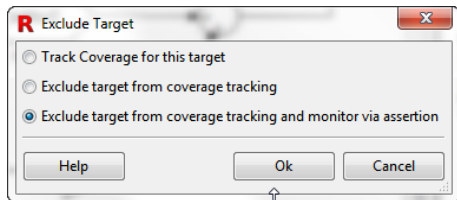
2) Right-click on block with target

3) Select *Track Coverage*, then select target to disable

4) Select exclusion status

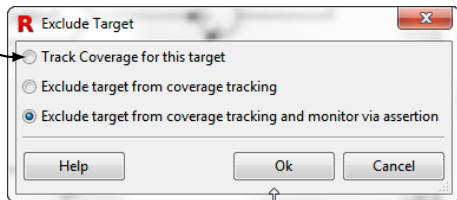


Exclusion Status Options



Exclusion Status Options

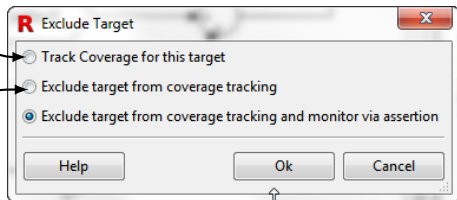
Track coverage
as usual



Exclusion Status Options

Track coverage
as usual

Ignore target

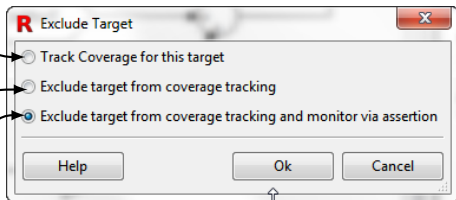


Exclusion Status Options

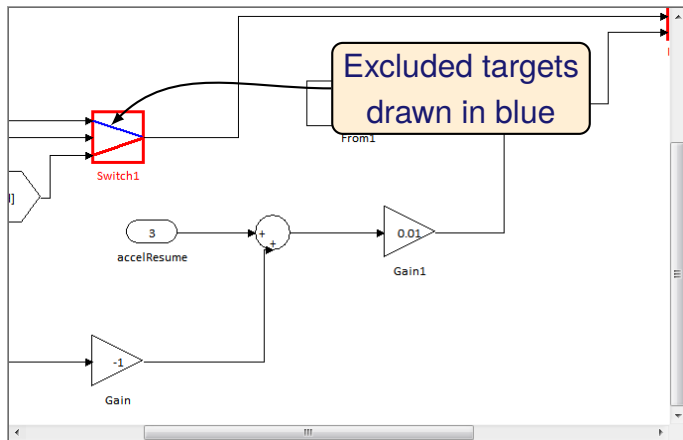
Track coverage
as usual

Ignore target

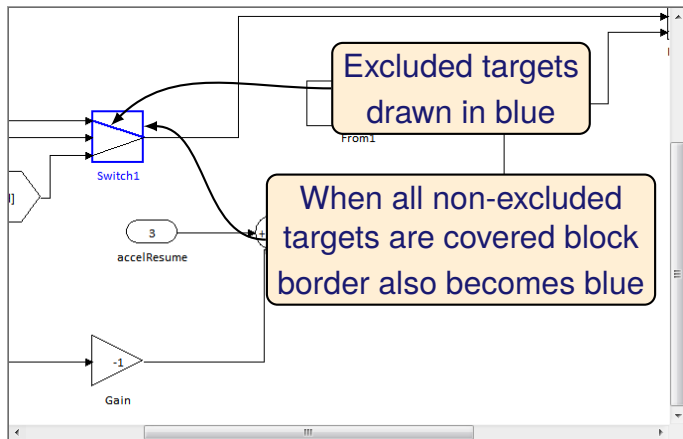
Ignore target, but
flag assertion violation
if target is exercised



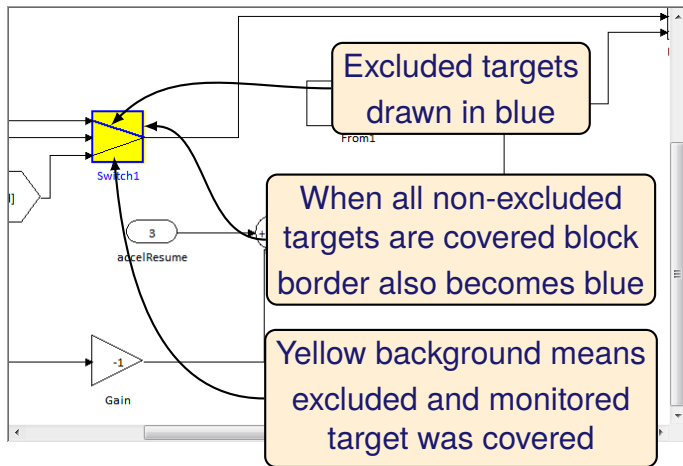
Visualization of Excluded Targets



Visualization of Excluded Targets

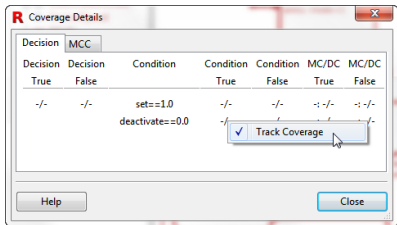


Visualization of Excluded Targets



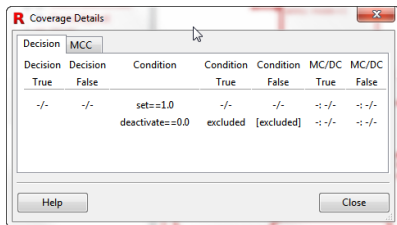
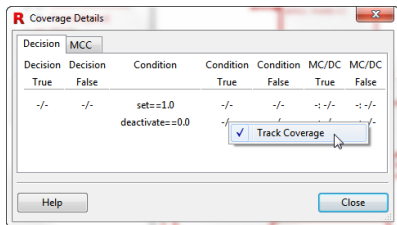
Excluding Targets from Coverage Details Dialog

Right-click on any test/step location and select *Track Coverage*



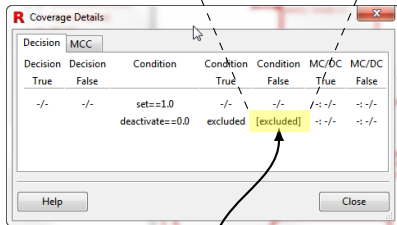
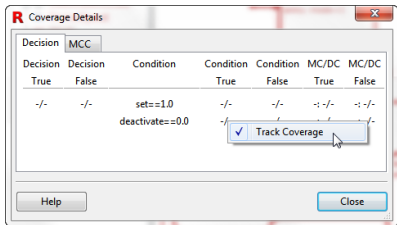
Excluding Targets from Coverage Details Dialog

Right-click on any test/step location and select *Track Coverage*



Excluding Targets from Coverage Details Dialog

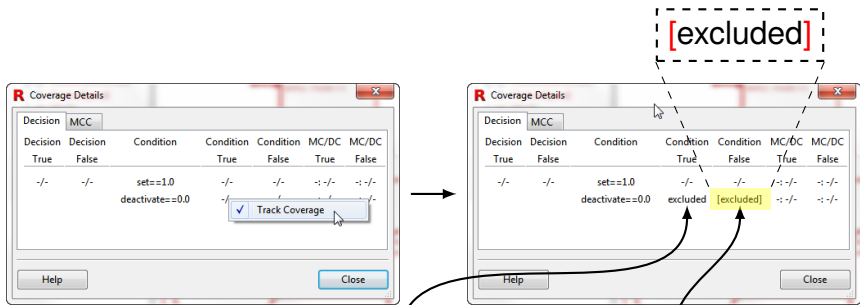
Right-click on any test/step location and select *Track Coverage*



Braces indicate excluded and monitored

Excluding Targets from Coverage Details Dialog

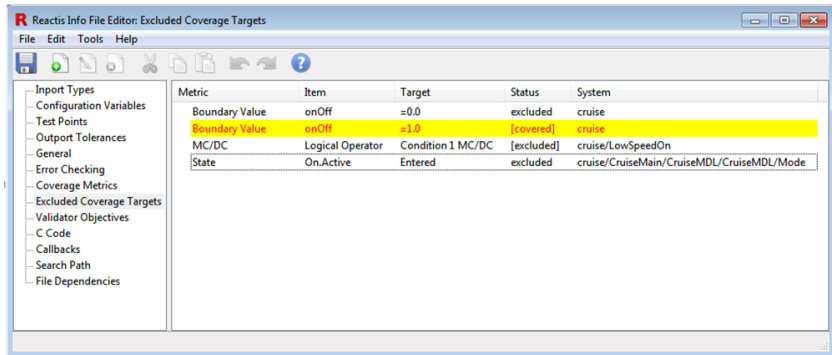
Right-click on any test/step location and select *Track Coverage*



No braces indicates excluded but not monitored

Braces indicate excluded and monitored

Excluded Targets Pane in Info File Editor



Reactis Info File Editor: Excluded Coverage Targets

File Edit Tools Help

Import Types
Configuration Variables
Test Points
Output Tolerances
General
Error Checking
Coverage Metrics
Excluded Coverage Targets
Validator Objectives
C Code
Callbacks
Search Path
File Dependencies

Metric	Item	Target	Status	System
Boundary Value	onOff	=0.0	excluded	cruise
Boundary Value	onOff	=1.0	[covered]	cruise
MC/DC	Logical Operator	Condition 1 MC/DC	[excluded]	cruise/LowSpeedOn
State	On.Active	Entered	excluded	cruise/CruiseMain/CruiseMDL/CruiseMDL/Mode

Reactis for EML Plugin

White-box testing of Embedded MATLAB (EML) portions of a model.

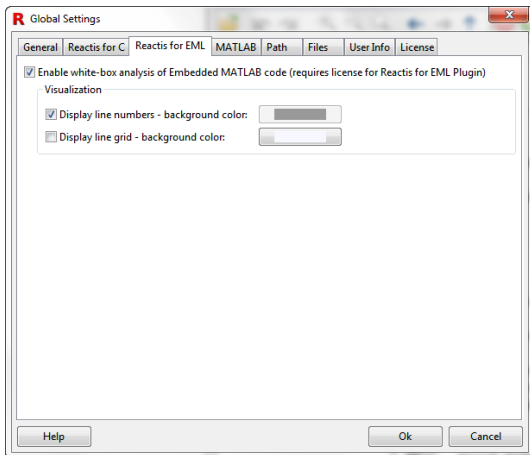
- ▶ Statement, decision, condition, MC/DC targets identified in EML code.
- ▶ Tester attempts to exercise EML targets.
- ▶ Simulator reports coverage of EML targets.

Enabling the EML Plugin

1) Select *File* → *Global Settings...*

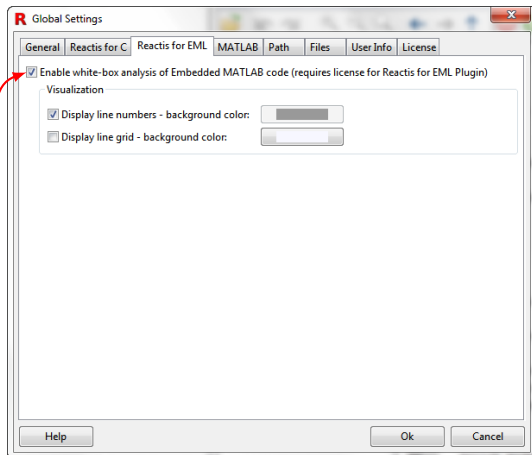
Enabling the EML Plugin

1) Select *File* → *Global Settings...*



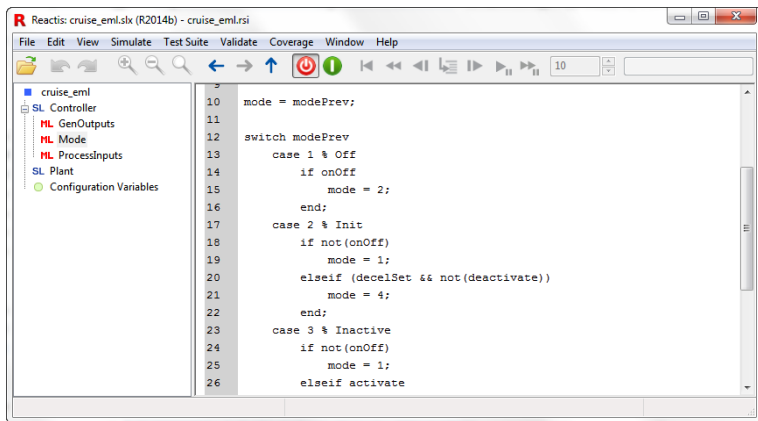
Enabling the EML Plugin

1) Select *File* → *Global Settings...*



2) Check
box

Display of EML Code



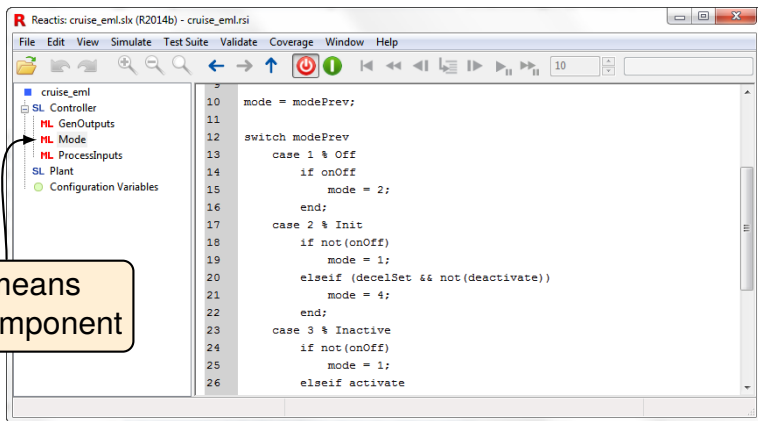
The screenshot shows the Reactis IDE interface for a project named "cruise_eml.slx (R2014b) - cruise_eml.rsi". The window title bar includes standard OS controls. The menu bar contains "File", "Edit", "View", "Simulate", "Test Suite", "Validate", "Coverage", "Window", and "Help". Below the menu bar is a toolbar with icons for file operations, search, navigation, and simulation control. A left-hand pane displays a project tree with the following structure:

- cruise_eml
 - SL Controller
 - ML GenOutputs
 - ML Mode
 - ML ProcessInputs
 - SL Plant
 - Configuration Variables

The main editor area displays EML code for a switch statement that controls the mode of a system based on various inputs. The code is as follows:

```
10 mode = modePrev;
11
12 switch modePrev
13     case 1 % Off
14         if onOff
15             mode = 2;
16         end;
17     case 2 % Init
18         if not(onOff)
19             mode = 1;
20         elseif (decelSet && not(deactivate))
21             mode = 4;
22         end;
23     case 3 % Inactive
24         if not(onOff)
25             mode = 1;
26         elseif activate
```


Display of EML Code



```
10 mode = modePrev;
11
12 switch modePrev
13     case 1 % Off
14         if onOff
15             mode = 2;
16         end;
17     case 2 % Init
18         if not(onOff)
19             mode = 1;
20         elseif (decelSet && not(deactivate))
21             mode = 4;
22         end;
23     case 3 % Inactive
24         if not(onOff)
25             mode = 1;
26         elseif activate
```

ML means
EML component

Display of EML Code

The screenshot shows the Reactis IDE interface for a project named 'cruise_eml'. The left sidebar contains a project tree with the following structure:

- cruise_eml
 - SL Controller
 - ML** GenOutputs
 - ML** Mode
 - ML** ProcessInputs
 - SL Plant
 - Configuration Variables

The main panel displays the EML code for the 'Mode' component, which is a switch statement:

```
10 mode = modePrev;  
11  
12 switch modePrev  
13   case 1 % Off  
14     if onOff  
15       mode = 2;  
16     end;  
17   case 2 % Init  
18     if not(onOff)  
19       mode = 1;  
20     elseif (decelSet && not(deactivate))  
21       mode = 4;  
22     end;  
23   case 3 % Inactive  
24     if not(onOff)  
25       mode = 1;  
26     elseif activate
```

Two callout boxes provide additional information:

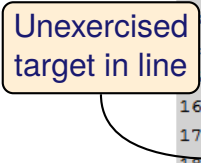
- A box on the left states: **ML** means EML component. An arrow points from this box to the 'ML' labels in the project tree.
- A box on the right states: EML code displayed in main panel. An arrow points from this box to the code in the main editor.

Tracking Coverage in EML Code in Simulator

```
12  switch modePrev
13      case 1 % Off
14          if onOff
15              mode = 2;
16          end;
17      case 2 % Init
18          if not(onOff)
19              mode = 1;
20          elseif (decelSet && not(deactivate))
21              mode = 4;
22          end;
23      case 3 % Inactive
24          if not(onOff)
25              mode = 1;
```

Tracking Coverage in EML Code in Simulator

Unexercised
target in line



```
12  switch modePrev
    case 1 % Off
        if onOff
            mode = 2;
        end;
    case 2 % Init
        if not(onOff)
            mode = 1;
        elseif (decelSet && not(deactivate))
            mode = 4;
        end;
    case 3 % Inactive
        if not(onOff)
            mode = 1;
```

Tracking Coverage in EML Code in Simulator

Unexercised
target in line

```
12  switch modePrev
    case 1 % Off
        if onOff
            mode = 2;
        end;
    case 2 % Init
        if not(onOff)
            mode = 1;
        elseif (decelSet && not(deactivate))
            mode = 4;
        end;
    case 3 % Inactive
        if not(onOff)
            mode = 1;
```

Statement
Coverage

Tracking Coverage in EML Code in Simulator

Unexercised
target in line

Decision
True

Statement
Coverage

```
12  switch modePrev
    case 1 % Off
        if onOff
            mode = 2;
        end;
16
17  case 2 % Init
    if not(onOff)
18      mode = 1;
19
    elseif (decelSet && not(deactivate))
        mode = 4;
    end;
23  case 3 % Inactive
    if not(onOff)
24      mode = 1;
25
```

Tracking Coverage in EML Code in Simulator

Unexercised target in line

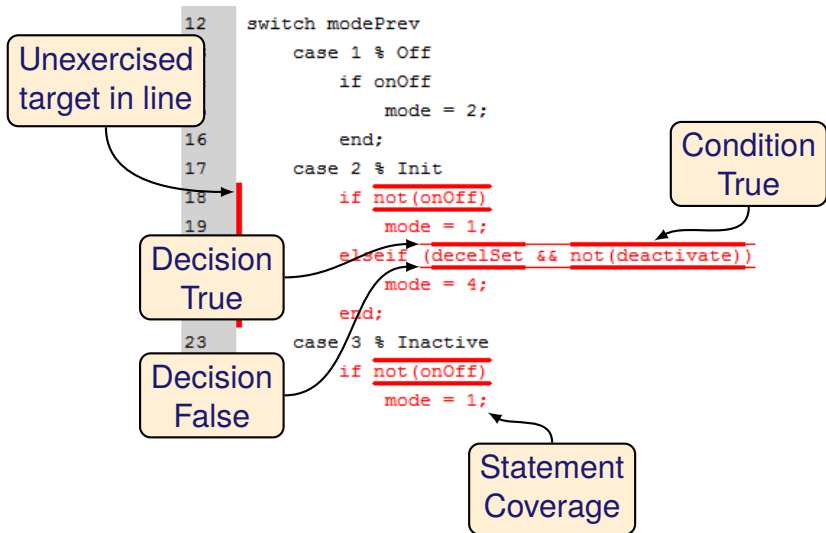
```
12  switch modePrev
    case 1 % Off
        if onOff
            mode = 2;
        end;
    case 2 % Init
        if not(onOff)
            mode = 1;
        elseif (decelSet && not(deactivate))
            mode = 4;
        end;
    case 3 % Inactive
        if not(onOff)
            mode = 1;
        end;
end;
```

Decision True

Decision False

Statement Coverage

Tracking Coverage in EML Code in Simulator



Tracking Coverage in EML Code in Simulator



Other Enhancements

- ▶ Track coverage in Direct Lookup Table blocks.
- ▶ API functions to add, remove, and modify configuration variables.

Thank You!

Reactis is a registered trademark of Reactive Systems, Inc.
MATLAB, Simulink, and Stateflow are registered trademarks of
MathWorks, Inc.