

What's New in Simulink

R2011a

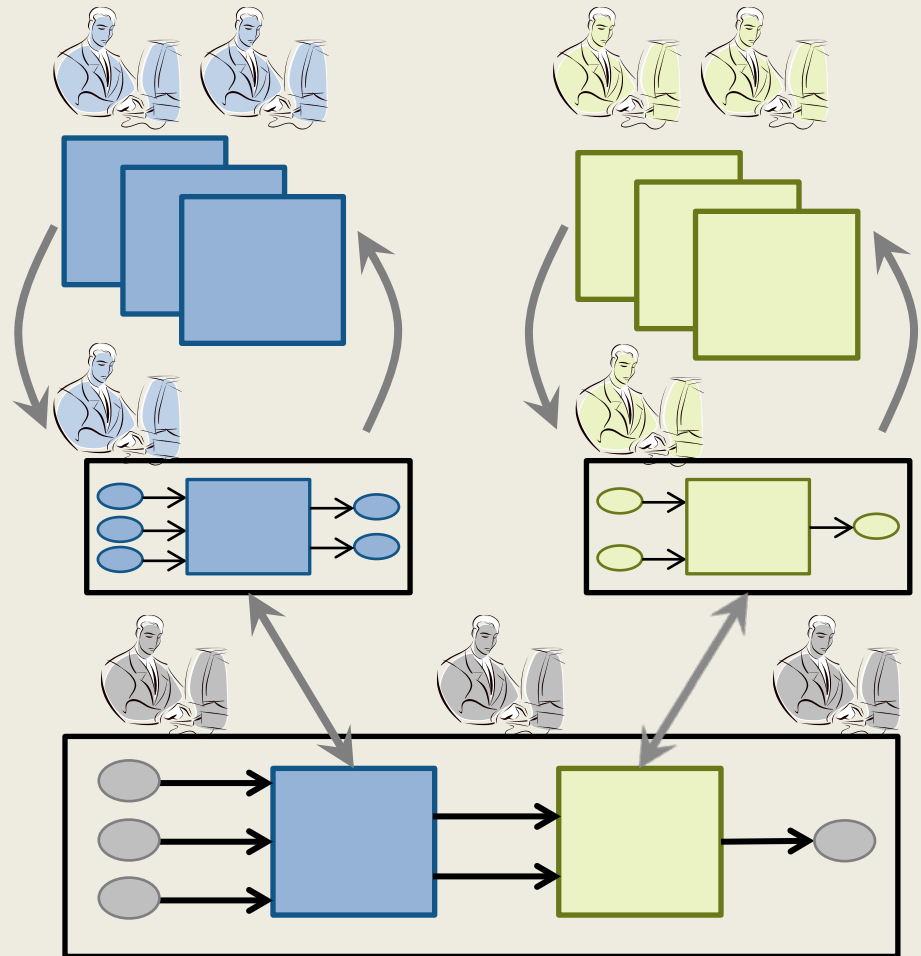
Themes

- Component-based modeling
- Data management
- Modeling language
- Performance
- User interface
- Project management

Why MathWorks Is Investing in Component-Based Modeling

Enables modular design for efficient and robust system development

- **Facilitate collaboration**, especially for modeling large systems
- **Partition** algorithm specifications, physical models, and tests
- Improve **iteration**, **verification**, and **configuration management**



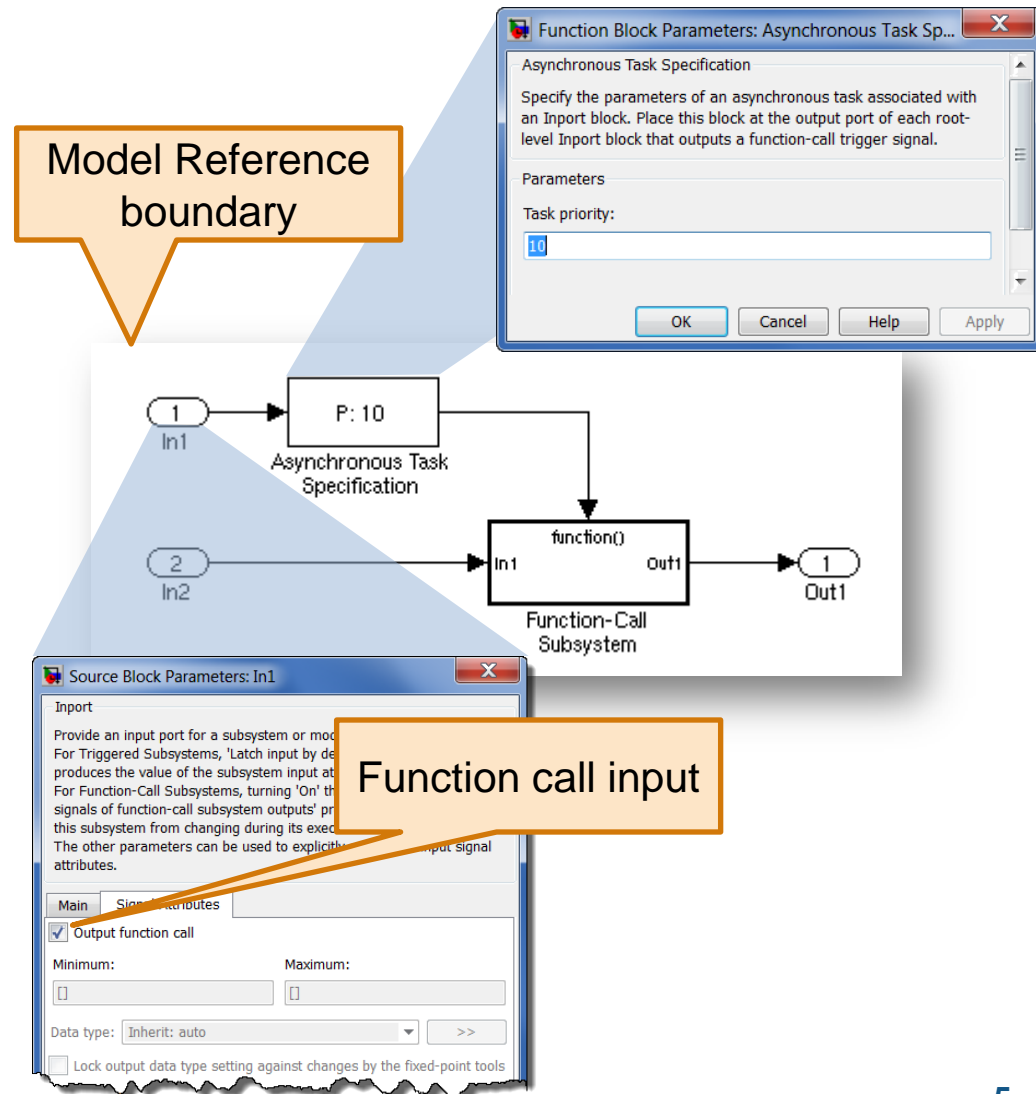
Component-Based Modeling

- Improve componentization of complex systems with asynchronous behavior
- Resolve data dependencies for interconnected function-call subsystems
- Manage data logging settings for large-scale models in a single GUI
- Track model names corresponding to all possible variants in one interface
- Protect models you share with just two clicks

Function-Call Input Support for Model References

Improve componentization of complex systems with asynchronous behavior

- Configure the model reference input as a function-call input
- Use the new Asynchronous Task Specification block to prioritize and route asynchronous events

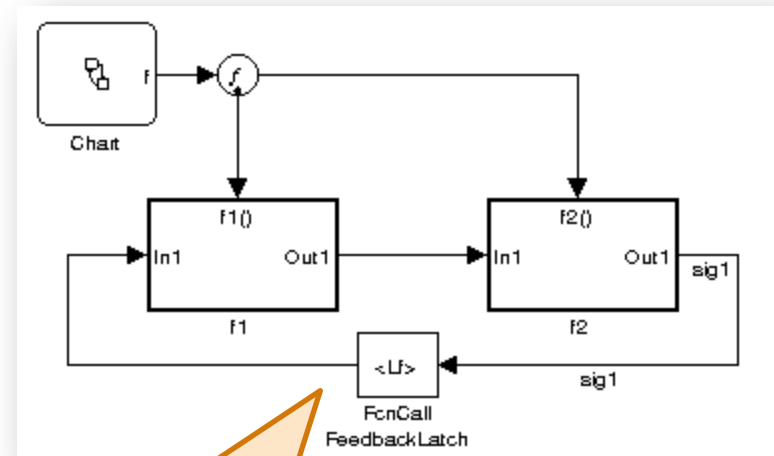


» sldemo_mdhref_fcncall

New Function-Call Feedback Latch Block

Resolve data dependencies for interconnected function-call subsystems

- Break feedback loop involving data signals between function-call blocks by introducing a delay in data signals
- Design destination function-call subsystem in a modular fashion and use it for creating in- or out-of-context loop scenarios

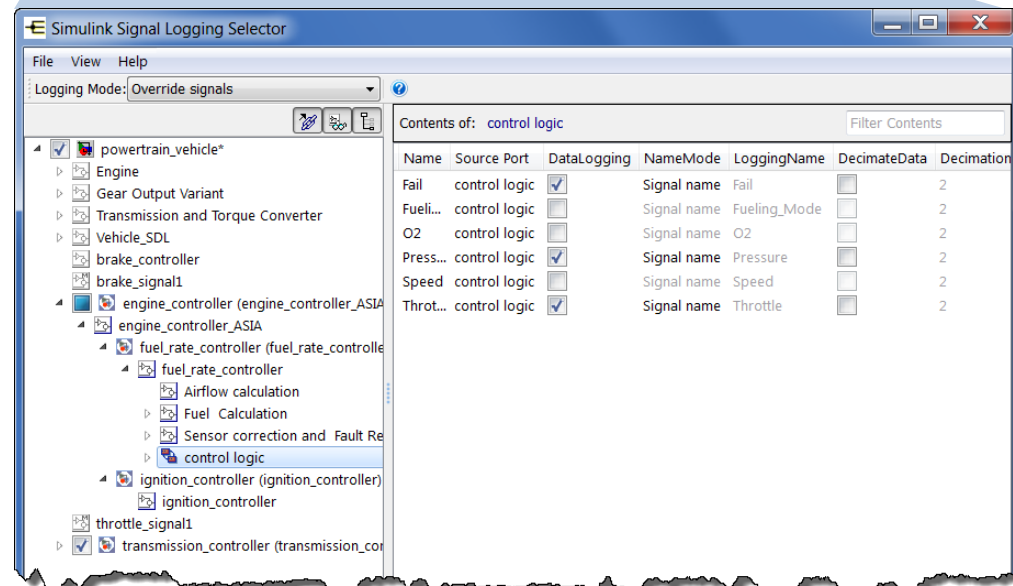
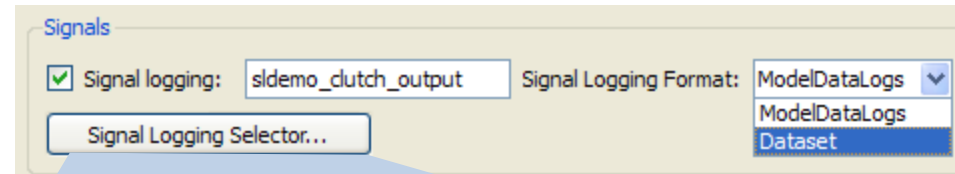


Breaks the dependency between $f1$ and $f2$

Signal Logging Selector GUI

Manage data logging settings for large-scale models in a single GUI

- Support for model reference and Stateflow charts
- Option to override model reference logging settings

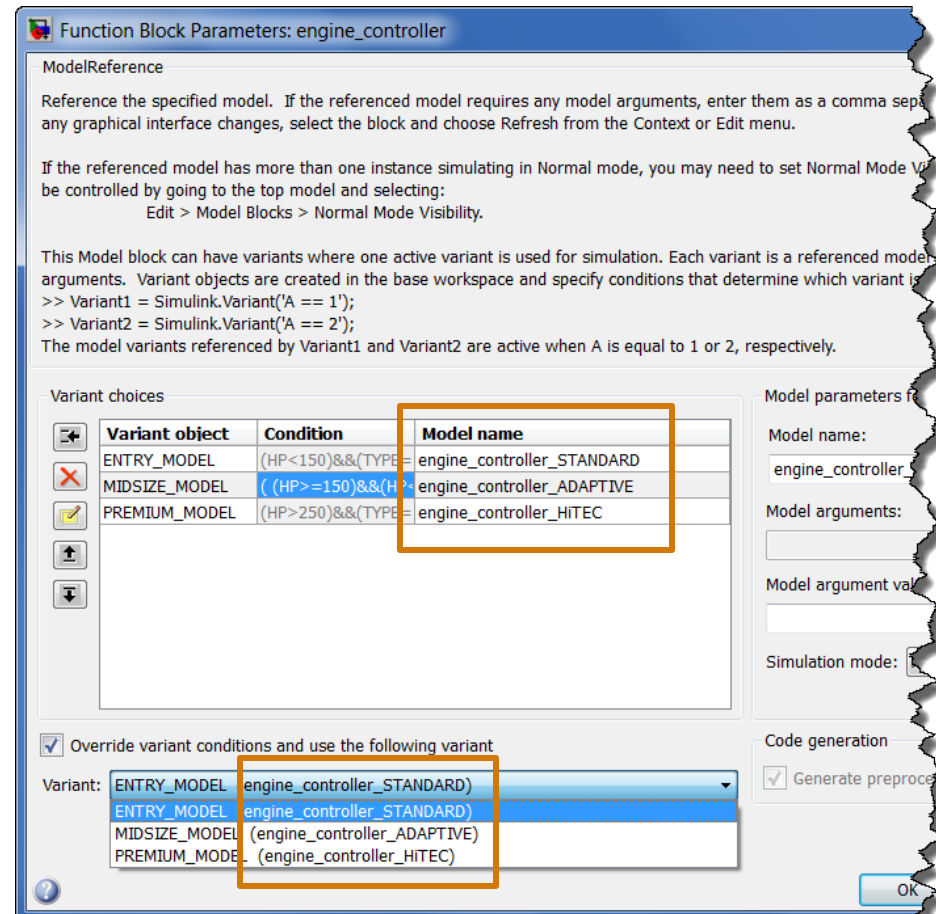


» sldemo_clutch

Better Variant Handling Through Enhanced User Interface

Track model names corresponding to all possible variants in one interface

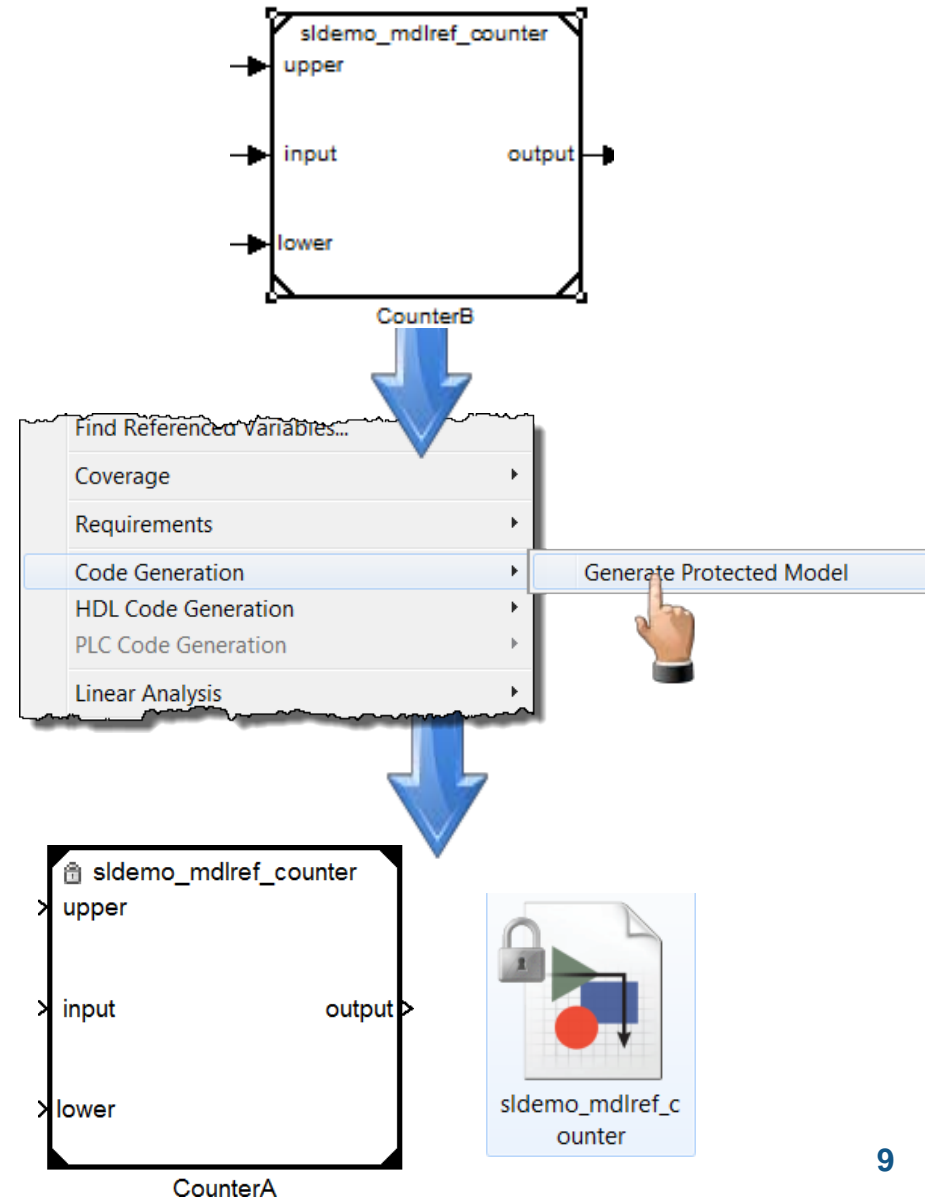
- Centralized dialog box includes both the conditions for variant choices and a mechanism to override these conditions.



Protected Mode Available as a Clickable Option

Protect models you share with just two clicks

- Create the protected model reference file with two clicks
- Share model functionality without exposing underlying implementation

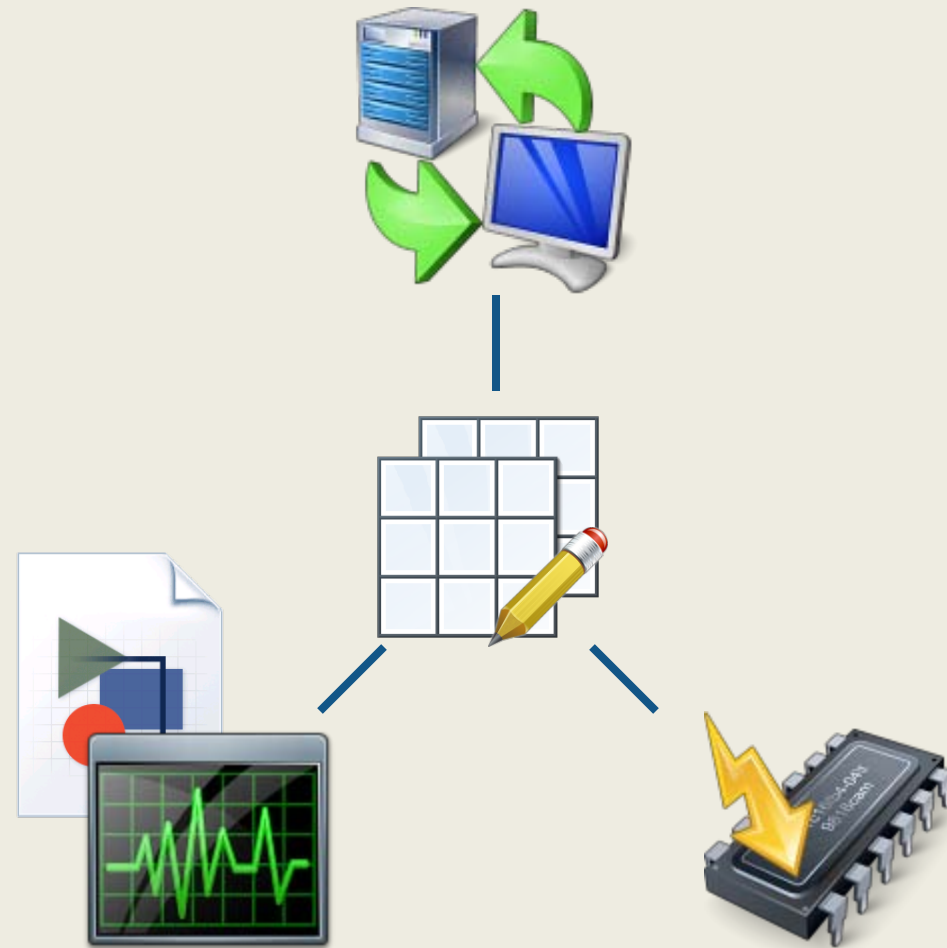


» sldemo_mdref_bus

Why MathWorks Is Investing in Data Management

Manage model data in all forms with ease

- Specify **data characteristics**
- Use **external data** for signal streaming, parameters, and test cases
- Implement computations with **fixed- and floating-point** types
- Identify and manage **data dependencies** of a model



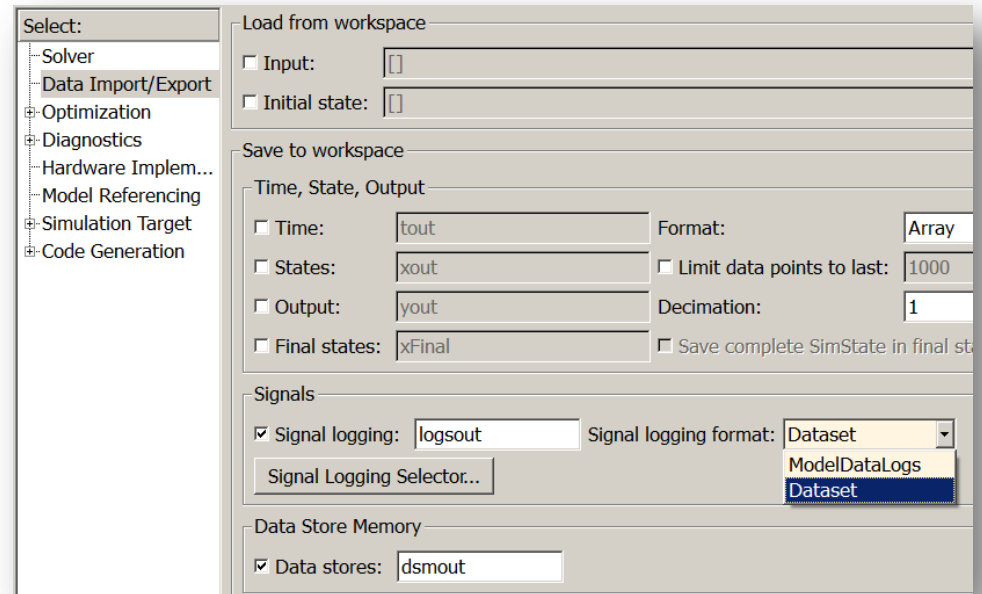
Data Management

- Efficiently log and analyze large signal data
- Parameterize the size of data objects
- Pack and access multiple signals efficiently
- Efficiently manage and route groups of signals as buses
- Customize your bus objects with MATLAB code

New Dataset Format for Signal Logging

Efficiently log and analyze large signal data

- Easily analyze logged data for models with deep hierarchies, bus signals, and signals with duplicate or invalid names
- Log multiple data values for a given time step when using an Iterator subsystem or Stateflow chart
- Use MATLAB `timeseries` functions for analysis and visualization

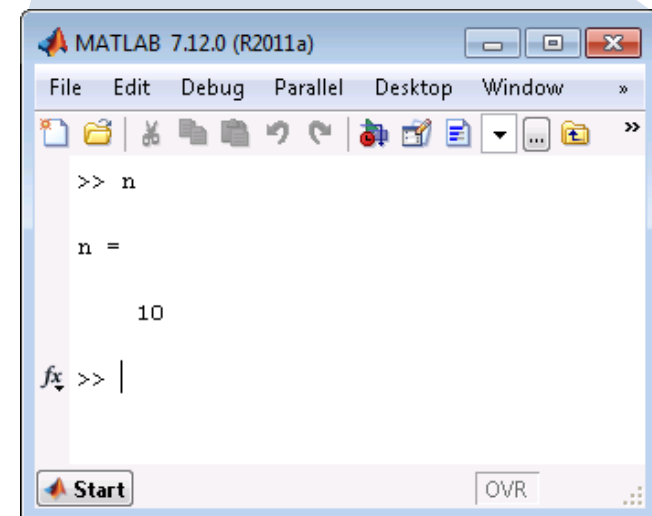
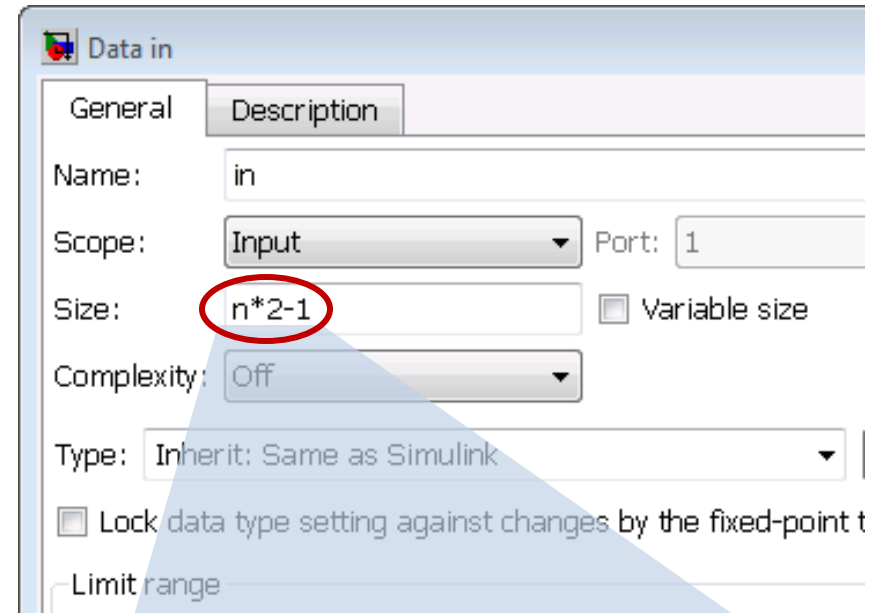


```
>>logout.getElement(1)
>>logout.getElement('sigName')
>>plot(logout.getElement(1).Values)
```

Use of MATLAB Expressions to Specify Data Size

Parameterize the size of data objects

- Data object size can be specified using any valid MATLAB expression that is evaluated at run time.
- Any data on the MATLAB path or in the workspace can be used.
- Data size no longer needs to be hardcoded.

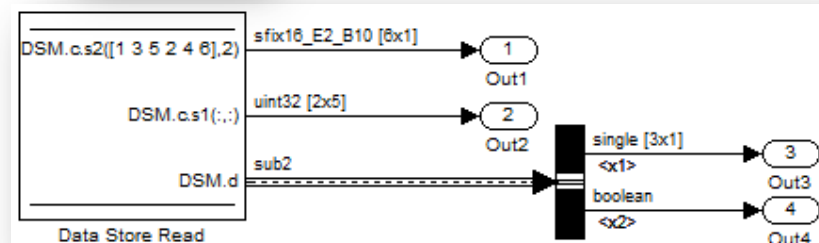
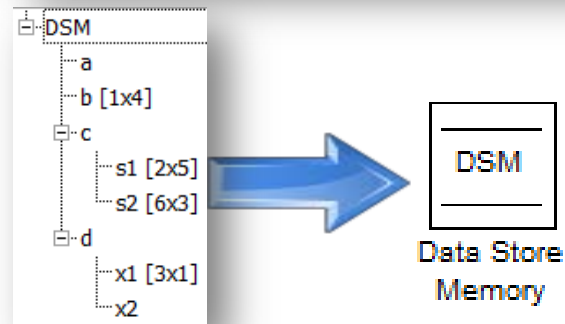
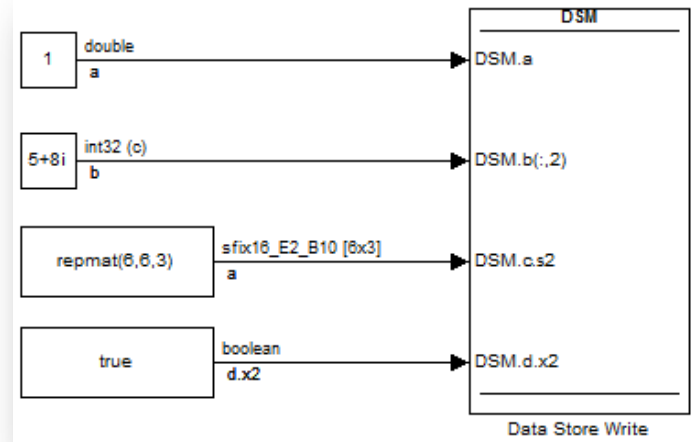


I/O Support for Buses in Data Store

Read, Write, and Memory Blocks

Pack and access multiple signals efficiently

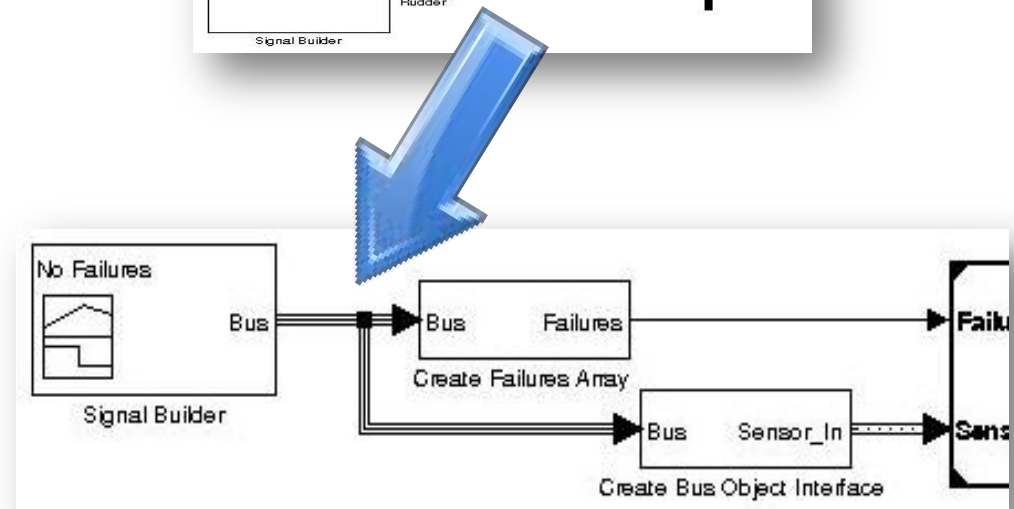
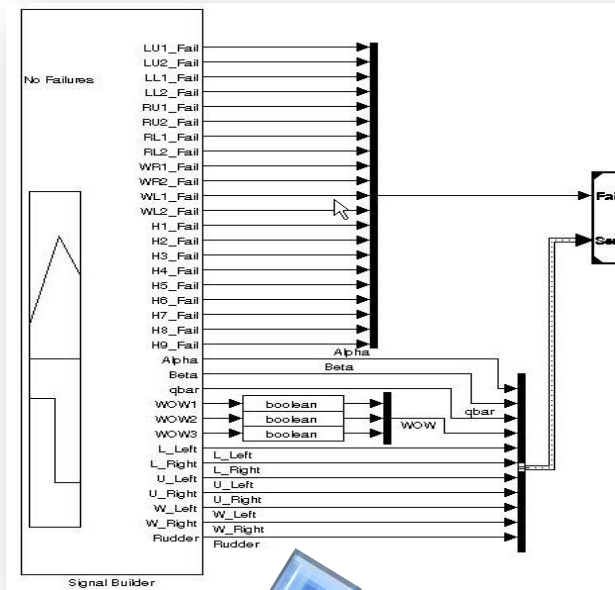
- Minimize the number of blocks used with efficient signal routing
- Generate C code for data store blocks as structures
- Save memory by avoiding redundant copies during I/O operations in Data Store Memory block



Bus Option for Signal Grouping in Signal Builder

Efficiently manage and route groups of signals as buses

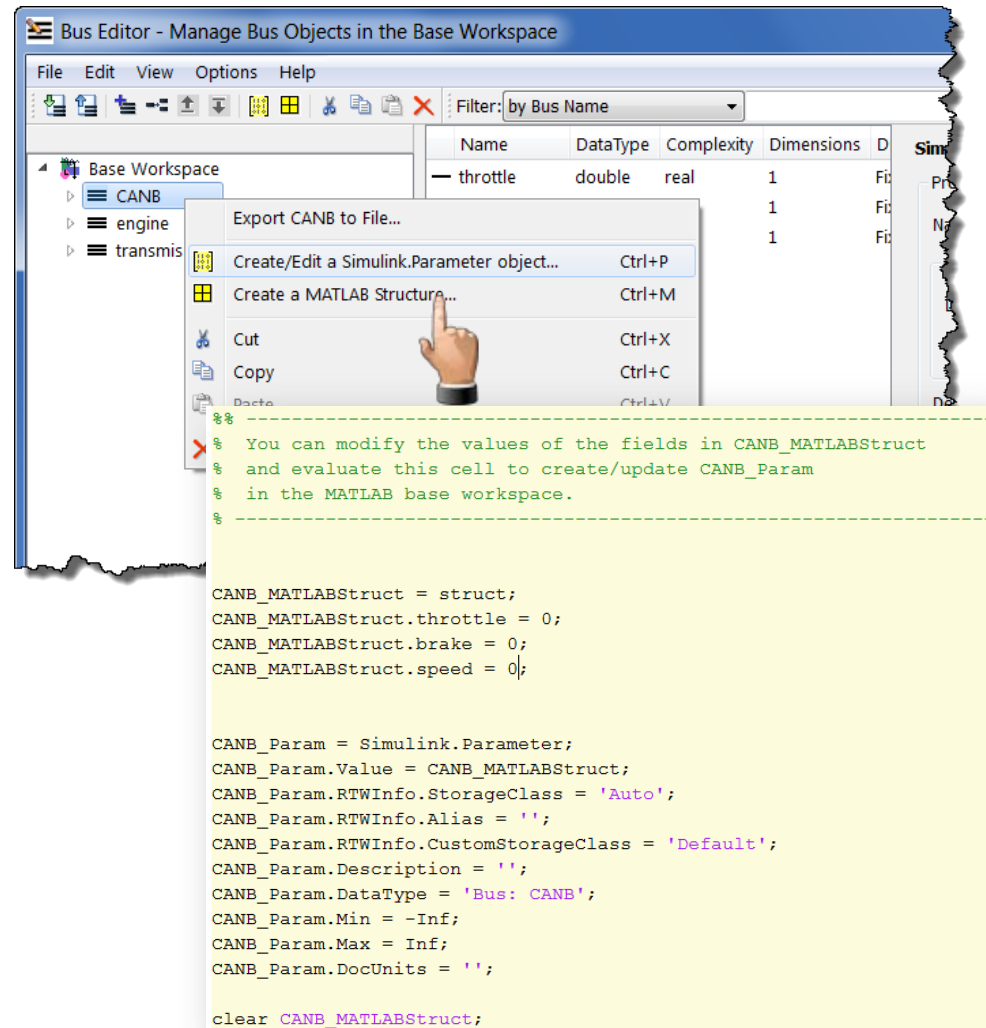
- Output a virtual or non-hierarchical bus signal
- Add or remove signals without rewiring
- Change the order of signals without graphically modifying the model



Textual Editing for Data Objects in Bus Editor

Customize your bus objects with MATLAB code

- Generate MATLAB code for bus objects defined in the GUI
- Create a `Simulink.Parameter` object or a MATLAB structure based on a bus object
- Evaluate or save the generated MATLAB code
- Modify the generated MATLAB code that represents the newly created object



Bus Editor - Manage Bus Objects in the Base Workspace

File Edit View Options Help

Filter: by Bus Name

Name	DataType	Complexity	Dimensions	D	Sim
throttle	double	real	1		Fi
			1		Fi
			1		Fi

Export CANB to File...

- Create/Edit a Simulink.Parameter object... Ctrl+P
- Create a MATLAB Structure... Ctrl+M
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V

```

%%
% You can modify the values of the fields in CANB_MATLABstruct
% and evaluate this cell to create/update CANB_Param
% in the MATLAB base workspace.
%
CANB_MATLABstruct = struct;
CANB_MATLABstruct.throttle = 0;
CANB_MATLABstruct.brake = 0;
CANB_MATLABstruct.speed = 0;

CANB_Param = Simulink.Parameter;
CANB_Param.Value = CANB_MATLABstruct;
CANB_Param.RTWInfo.StorageClass = 'Auto';
CANB_Param.RTWInfo.Alias = '';
CANB_Param.RTWInfo.CustomStorageClass = 'Default';
CANB_Param.Description = '';
CANB_Param.DataType = 'Bus: CANB';
CANB_Param.Min = -Inf;
CANB_Param.Max = Inf;
CANB_Param.DocUnits = '';

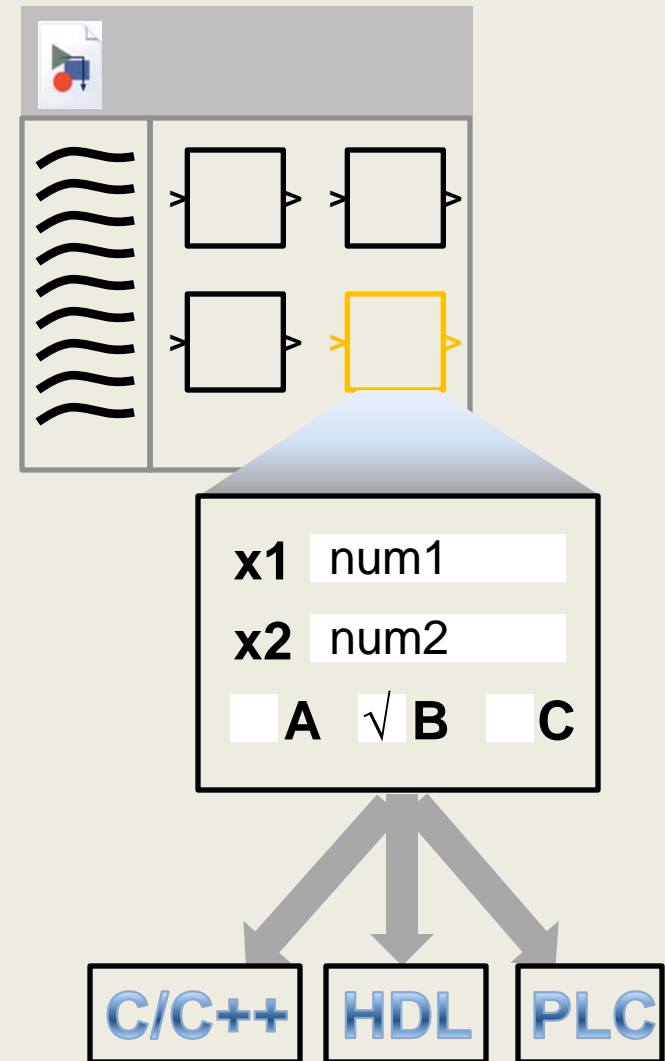
clear CANB_MATLABstruct;
  
```

» busdemo

Why MathWorks Is Investing in the Modeling Language

Enrich the Simulink modeling language to make it easier to express more ideas

- Increase the **scope** of the Simulink modeling language to expand what problems it can be used to solve
- Improve the **efficiency** of the Simulink modeling language and generated code
- Increase **control** over design tradeoffs and look of generate code



Modeling Language

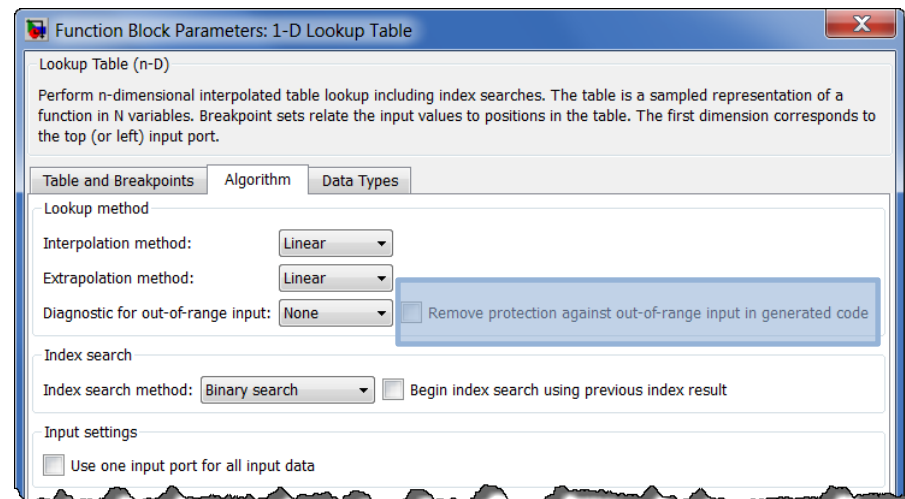
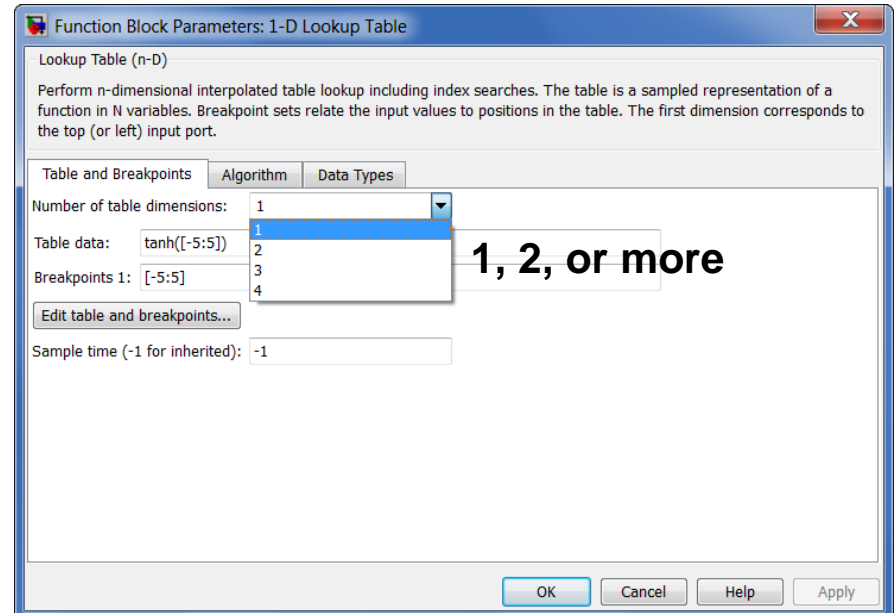
- Specify properties via common interface and generate smaller-footprint C code
- Change the bit shift using a signal input
- Use complex data, especially important for signal processing, in the Sign block

Unified Interface for Enhanced Lookup Tables

Specify properties via common interface and generate smaller footprint C code

- Use the same shared interface for 1-D, 2-D, and n-D Lookup Table blocks*
- Use n-D Lookup Table properties in 1-D and 2-D blocks
- Disable out-of-range protection diagnostic for smaller C code size
 - Prelookup block support

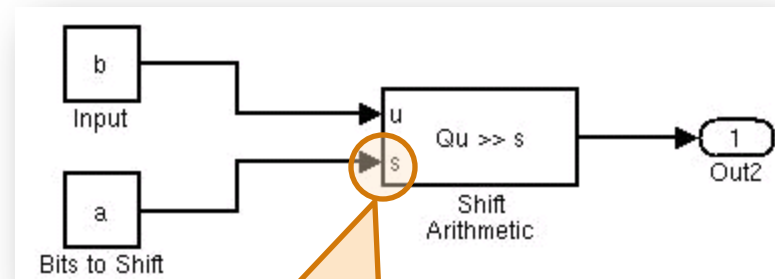
* Excludes Direct Lookup (n-D) Table



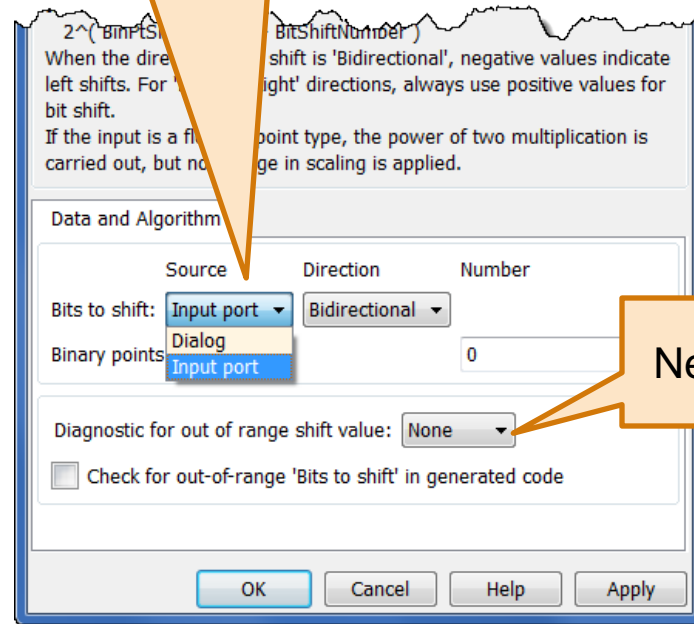
Variable Bit Shift Based on Signal Input

Change the bit shift using a signal input

- New diagnostic checks for out-of-range shift values in simulation and generated C code



Bits to shift as signal



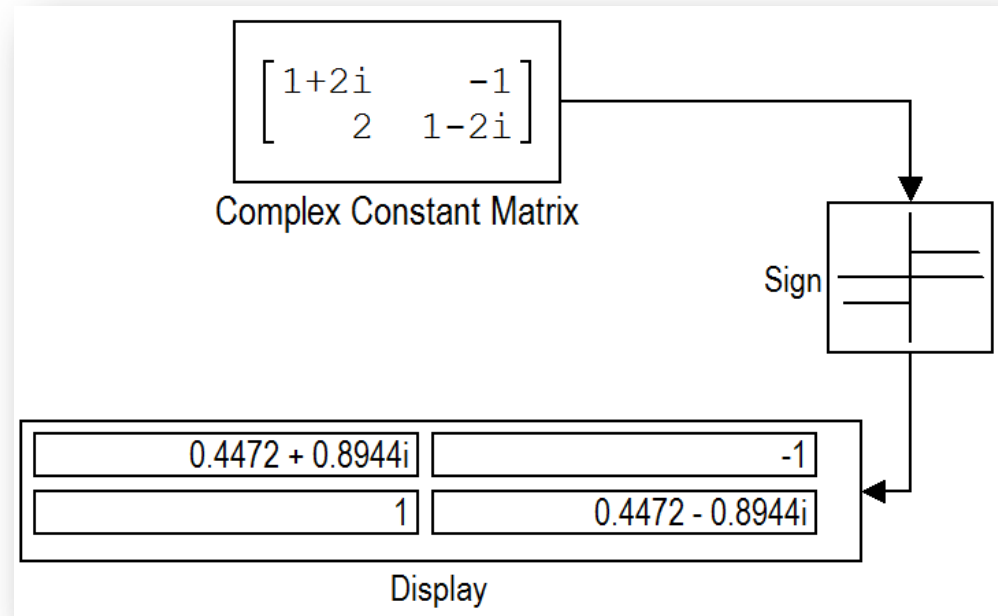
New diagnostic

Complex Floating-Point Support in Sign Block

Use complex data, especially important for signal processing, in the Sign block

- The Sign block can be used to determine sign for the real quantities for complex number-based computations.
- For a complex floating-point input scalar u , the output is:

$$\text{sign}(u) = u ./ \text{abs}(u)$$
- For a vector or matrix of complex numbers, each scalar value gets output based on its own complexity.

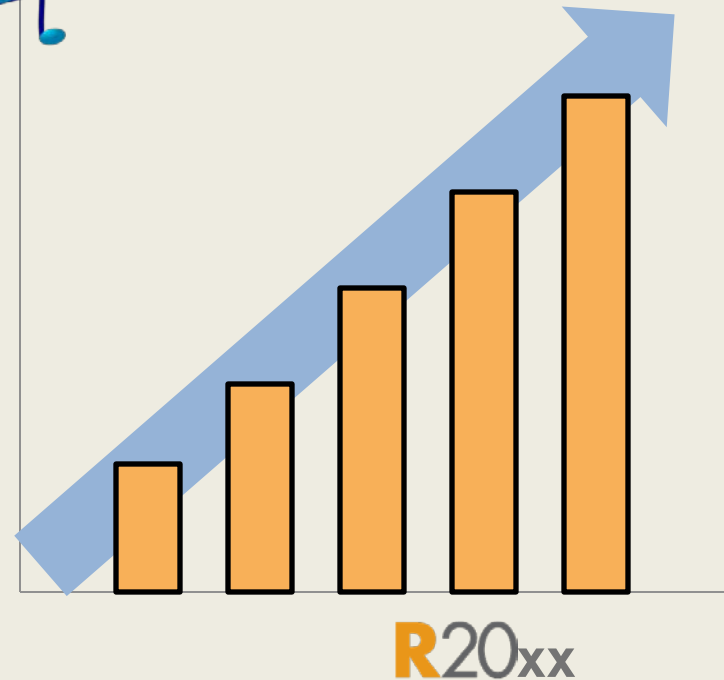


Why MathWorks Is Investing in Performance

Reduce the amount of time it takes to run simulations, test designs, and generate code

- Improve **simulation speed** to enable faster testing
- Improve **interactivity during simulation** to explore a larger design space
- Improve **code generation speed** for faster rapid prototyping, PIL/SIL/HIL testing, and production

speedup



Performance

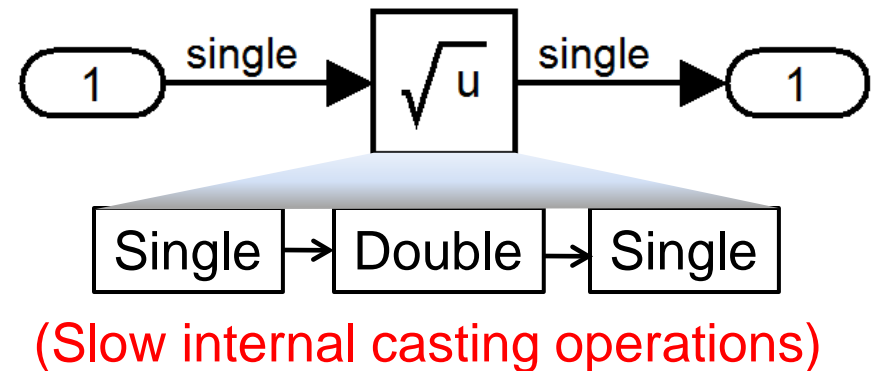
- Simulate faster single-precision computations with higher fidelity
- Continue simulations from any time point and from any Simulink release

Faster Single-Precision Computation

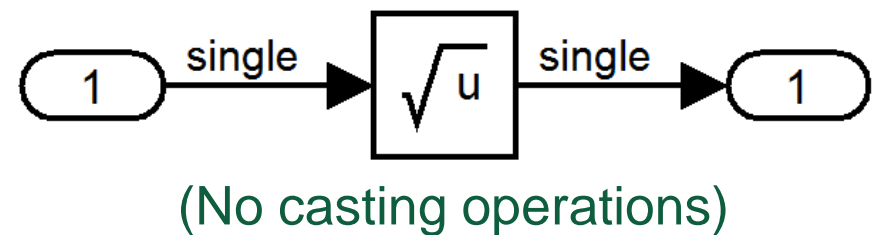
Simulate faster single-precision computations with higher fidelity

- Native single-precision implementation for elementary math operators
- Increased consistency between simulation and deployed code for single-precision computations
- Improved speed for simulation targets because of the elimination of up-casts and down-casts

Prior Releases



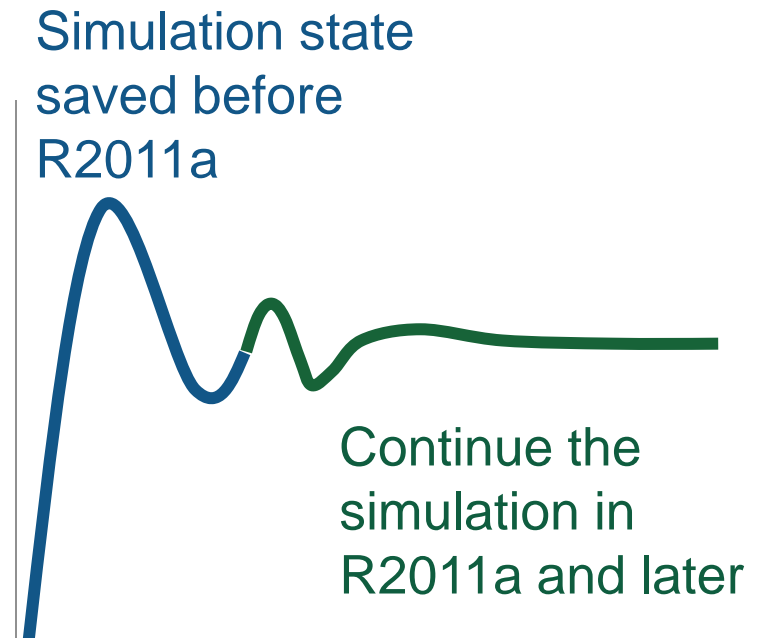
Today



Support for SimState from Previous Releases

Continue simulations from any time point and from any Simulink release

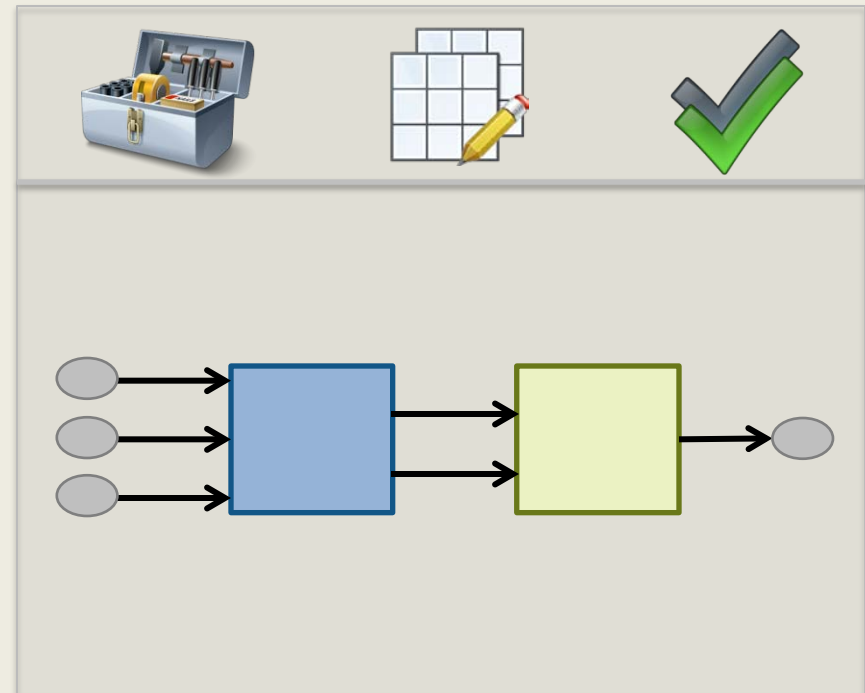
- The SimState feature was introduced in R2009a to reduce repeating long simulations by starting simulations from a saved state.
- This SimState data can now be loaded from models saved in prior releases of Simulink.



Why MathWorks Is Investing in User Interface

Spend more time designing and less time building and configuring models

- Quickly create **executable specifications**
- Access **analysis tools** without leaving the model editor
- **Manage data and model hierarchy** at a single place
- **Verify designs** using modeling standards



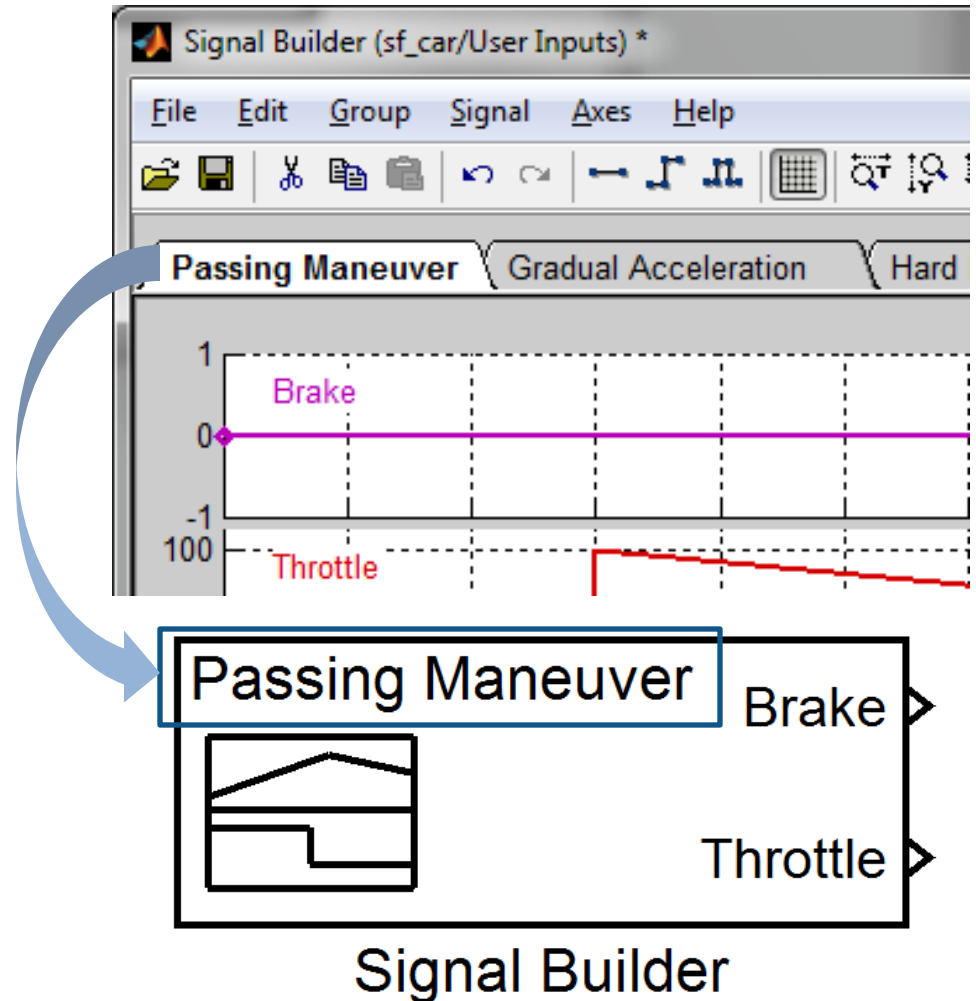
User Interface

- Easily identify the active group in the Signal Builder block
- Reuse the same Model Explorer tab view when inspecting or editing data

Active Group Block Mask Display Annotation

Easily identify the active group in the Signal Builder block

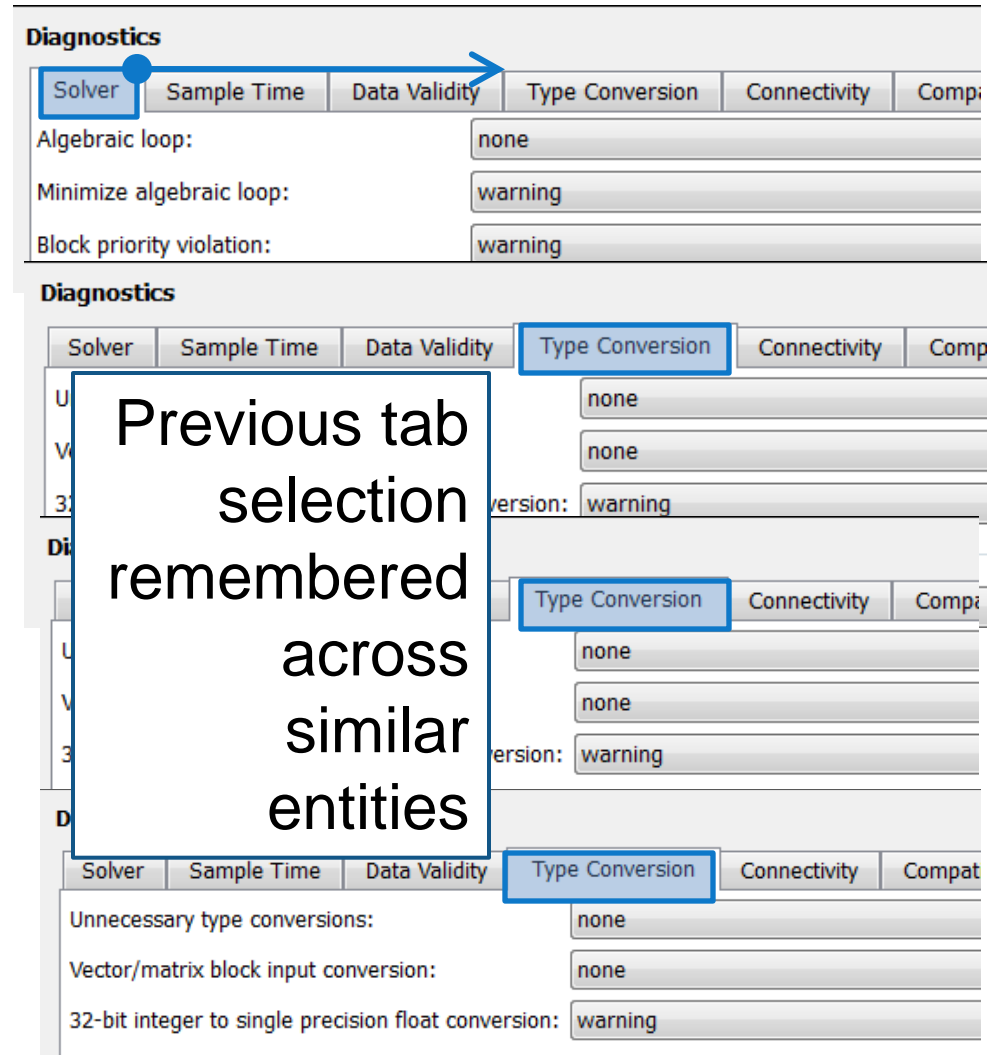
- The active group for a Signal Builder block now appears on the block within the Simulink editor.
- It is no longer necessary to open the block in order to check which group is currently active.



Sticky Tabs in Model Explorer GUI

Reuse the same Model Explorer tab view when inspecting or editing data

- Selected tab Model Explorer embedded dialog pane is remembered for tabbed dialog boxes.
- There is no need to reselect a tab when editing objects of similar types.

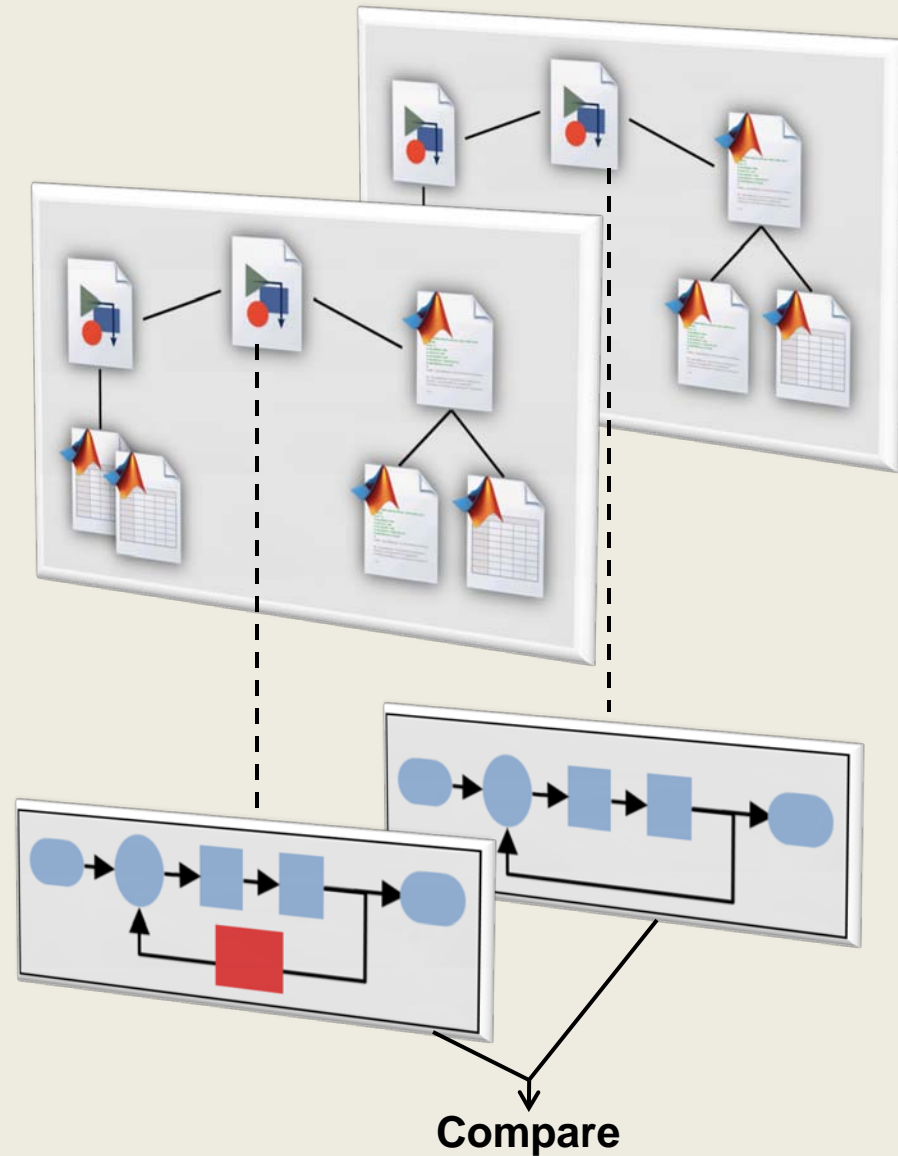


The image shows three sequential screenshots of the 'Diagnostics' dialog box in the Model Explorer GUI. In each screenshot, the 'Type Conversion' tab is selected and highlighted with a blue border. A blue arrow in the first screenshot points from the 'Solver' tab to the 'Type Conversion' tab. A text box in the center of the three screenshots reads: 'Previous tab selection remembered across similar entities'. The dialog box content changes based on the selected tab, showing various diagnostic categories and their status (e.g., 'Algebraic loop: none', 'Minimize algebraic loop: warning', 'Block priority violation: warning').

Why MathWorks Is Investing in Project Management

Improve how design components are shared and managed throughout a project's life cycle

- **Identify dependencies** for distributing and recreating designs
- **Compare versions** to track changes as the design evolves



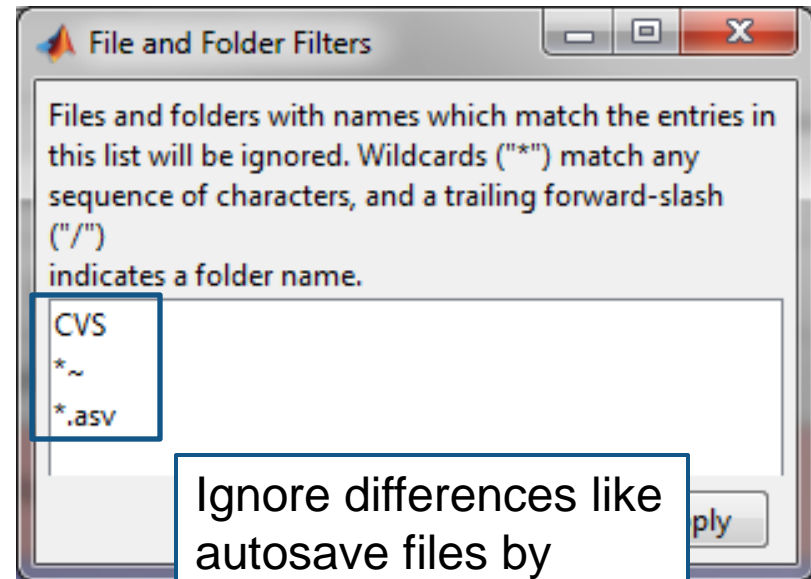
Project Management

- Compare only what matters and ignore obvious differences
- Merge Simulink models from within XML comparison report
- Merge differences in MAT-files and compare them in more detail

Improved Folder Comparison

Compare only what matters and ignore obvious differences

- Choose to ignore folder differences due to file types or configuration management folders
- Add filter items using a GUI
- See which items were ignored in the comparison report

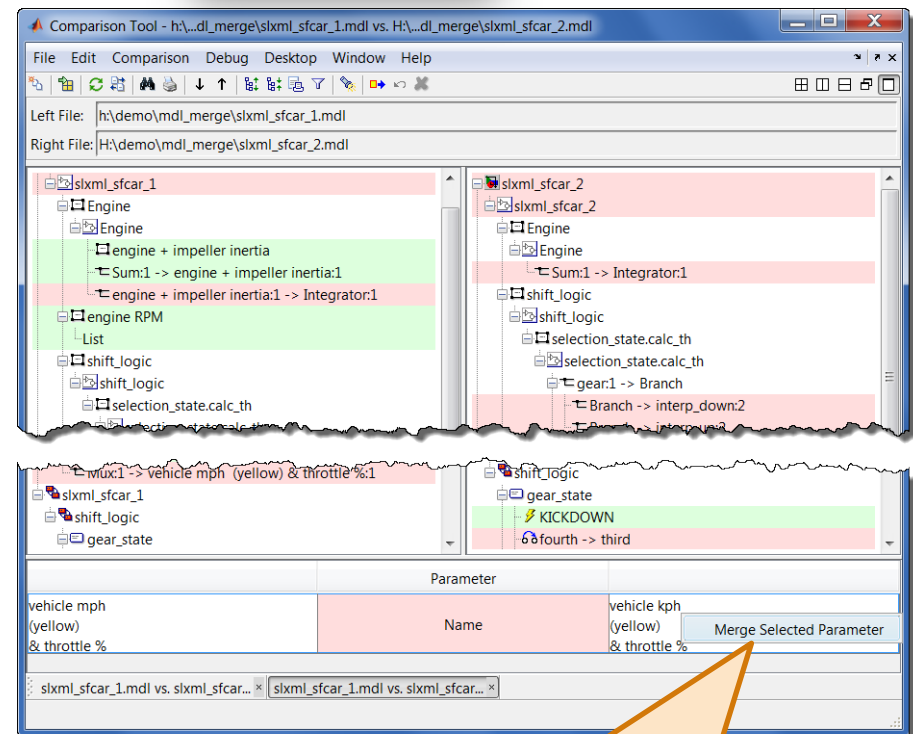


Type ▾		Change Summary
ASV file	enum_data.asv	<i>ignored</i>

Merge Simulink Models Based on XML Comparison Differences

Merge Simulink models from within XML comparison report

- Merge models within the tool by merging changes from left to right:
 - Left model is the base
 - Right model is the one edited
- Merge individual parameters, blocks, or entire subsystems



Parameter merge option for the selected node

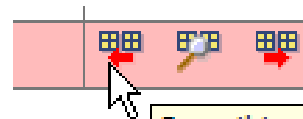
» slxml_sfcar

Merge MAT-Files with Detailed XML Comparison

Merge differences in MAT-files and compare them in more detail

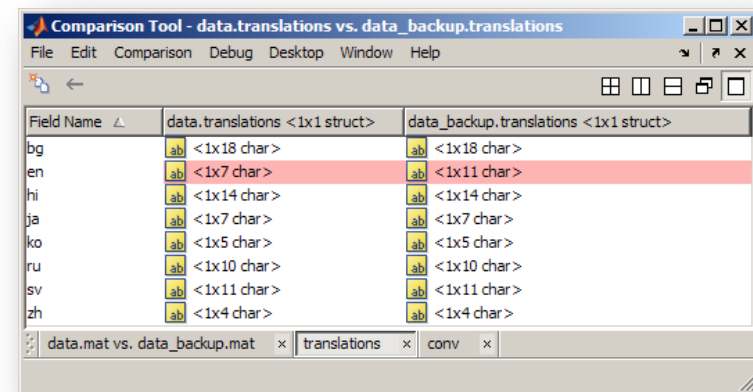
- Copy a variable from one file to another with a single mouse-click
- See which fields of a structure are different
- View differences in individual elements of an array

New Merge option



Copy this variable from the right-hand file to the left-hand file

Compare structure elements



Compare array elements

