

# Simulink 7.1

**R2008a**

# Agenda

- Simulink
  - **Simulink Library Browser Enhancements**
  - Simulation Performance
  - Component-Based Modeling
  - Embedded MATLAB
  - Data Management
  - File Management
  - Block Enhancements
  - User Interface Enhancements

# Platform Independence

## Challenge

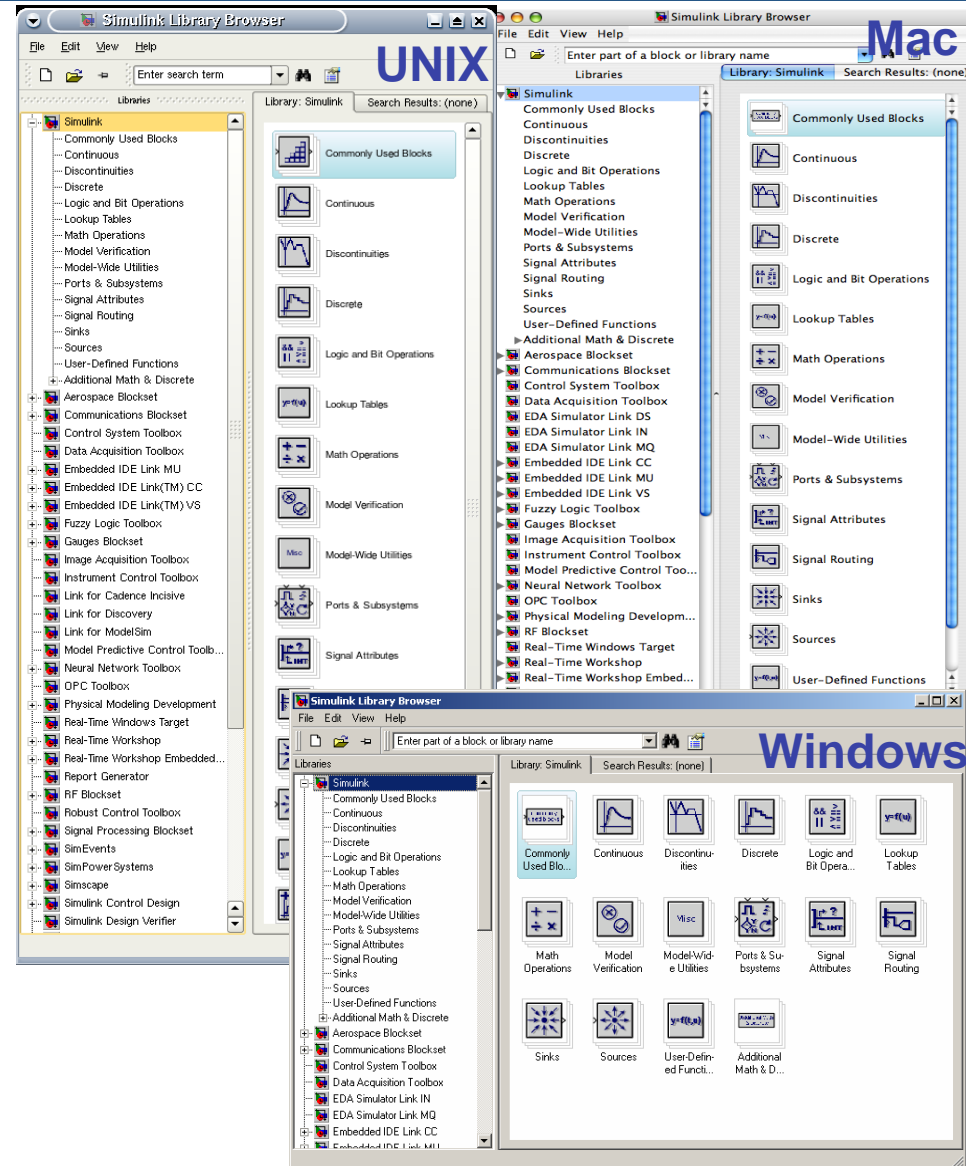
- Simulink Library Browser was a Windows only feature.
- Browsing large libraries or lots of libraries could be slow.

## Solution

- The Library Browser is now platform-independent and runs on Windows, Linux, Mac, and Solaris.
- Browsing no longer requires loading libraries into memory.

## Benefit

- Library Browser now available on all Simulink supported platforms.
- Browsing libraries is much faster.



# Search Enhancements

## Challenge

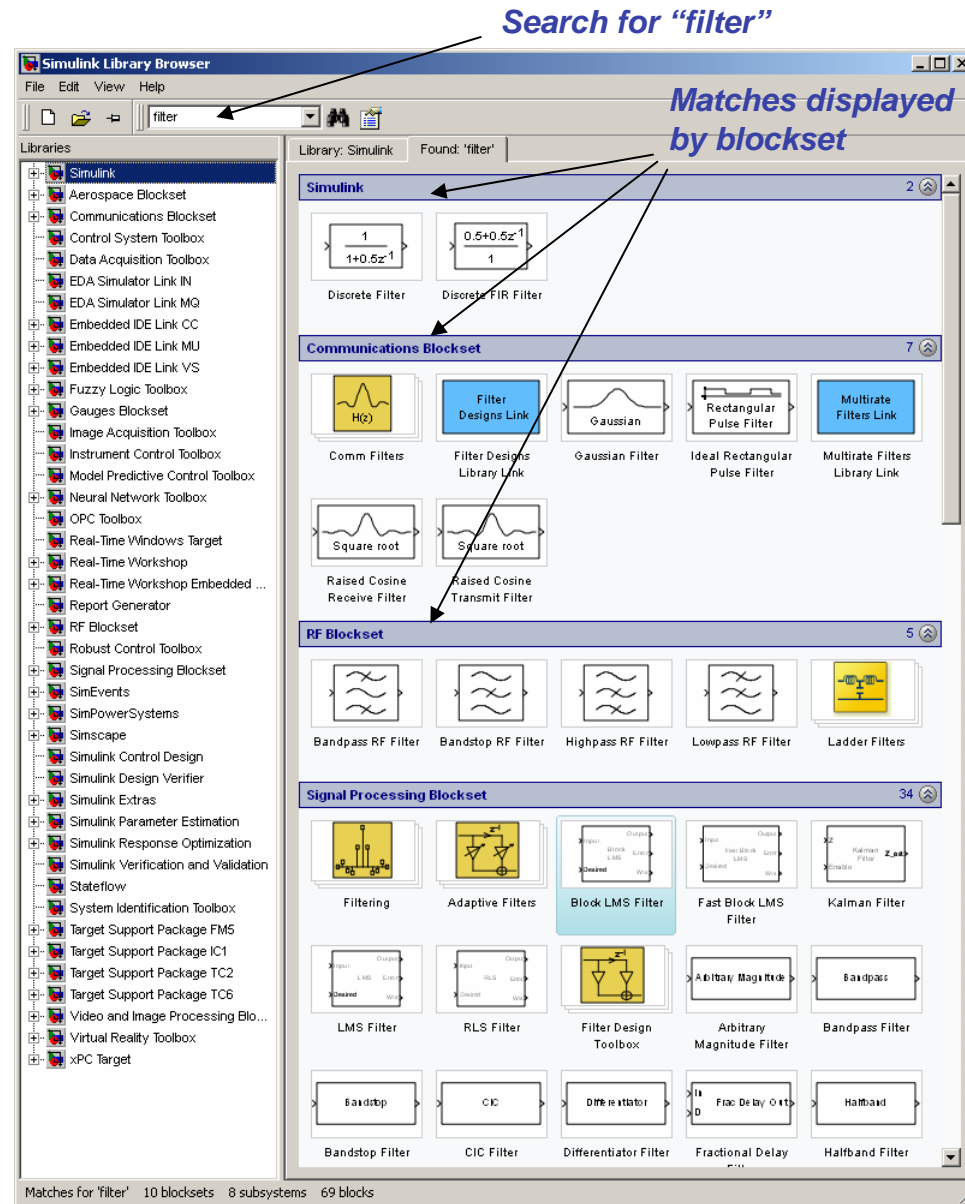
- Searching could be slow.
- Search was incremental, making it hard to find an item with a common name that has many matches (e.g., “filter”).

## Solution

- Browsing no longer requires loading libraries into memory.
- You can organize search results by blockset.
- New grid view shows results more compactly.

## Benefit

- Searching large libraries or lots of libraries is much faster.
- Search results are easier to review and navigate.



The screenshot shows the Simulink Library Browser interface. At the top, a search bar contains the text "filter". A label "Search for 'filter'" points to this search bar. Below the search bar, the results are organized into blocksets. A label "Matches displayed by blockset" points to the blockset headers. The blocksets shown are:

- Simulink** (2 matches): Discrete Filter, Discrete FIR Filter.
- Communications Blockset** (7 matches): Comm Filters, Filter Designs Library Link, Gaussian Filter, Ideal Rectangular Pulse Filter, Multirate Filters Library Link, Raised Cosine Receive Filter, Raised Cosine Transmit Filter.
- RF Blockset** (5 matches): Bandpass RF Filter, Bandstop RF Filter, Highpass RF Filter, Lowpass RF Filter, Ladder Filters.
- Signal Processing Blockset** (34 matches): Filtering, Adaptive Filters, Block LMS Filter, Fast Block LMS Filter, Kalman Filter, LMS Filter, RLS Filter, Filter Design Toolbox, Arbitrary Magnitude Filter, Bandpass Filter, Bandstop Filter, CIC Filter, Differentiator Filter, Fractional Delay Filter, Halfband Filter.

At the bottom of the window, a status bar indicates: "Matches for 'filter': 10 blocksets 8 subsystems 69 blocks".

# Agenda

- Simulink
  - Simulink Library Browser Enhancements
  - **Simulation Performance**
  - Component-Based Modeling
  - Embedded MATLAB
  - Data Management
  - File Management
  - Block Enhancements
  - User Interface Enhancements

# Adaptive Zero Crossing and Shape Preservation Integration

## Challenge

- Excessive zero crossing detections dramatically reduced simulation performance and affected simulation results.

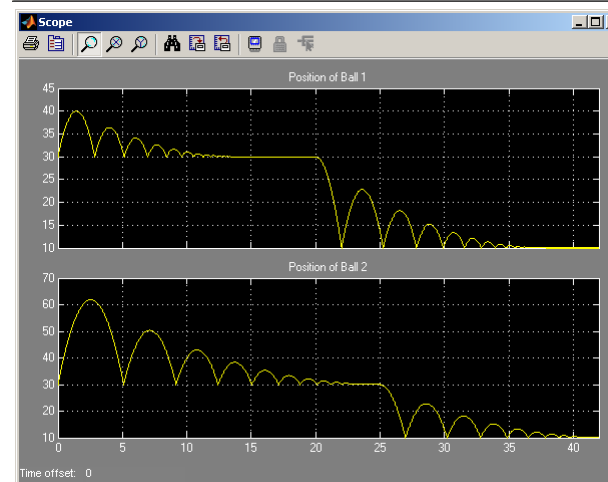
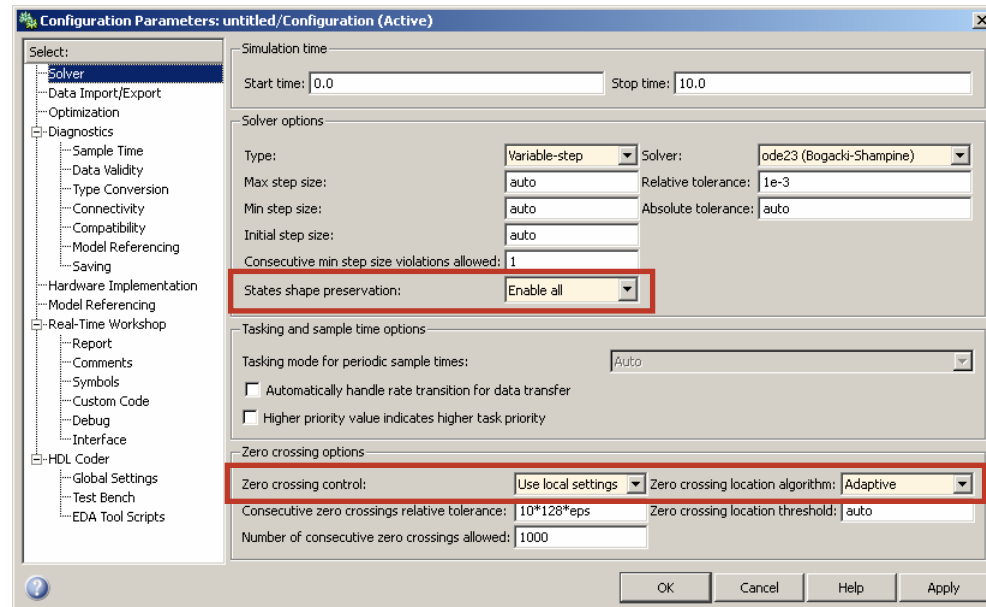
## Solution

- Simulink includes a new adaptive zero crossing location algorithm and state shape preservation algorithm that are compatible with all variable step solvers.

## Benefit

- Simulation speed is improved for models with a lot of zero crossings.
- Simulation accuracy is increased for models with high rates of change.
- Hybrid system simulation is improved.

>>sldemo\_doublebounce



# Agenda

- Simulink
  - Simulink Library Browser Enhancements
  - Simulation Performance
  - **Component-Based Modeling**
  - Embedded MATLAB
  - Data Management
  - File Management
  - Block Enhancements
  - User Interface Enhancements

# Model Reference Build Limitation Removed

## Challenge

- Compile-time issues can occur on a Windows host when a large number of models exist in a model reference hierarchy.
  - e.g., 1K model blocks, 200K total blocks

## Solution

- Ability to use a single include path and a response file for listing libraries

## Benefit

- Long command lines are no longer needed in the compilation process.
- Scalability is no longer limited by host platform restrictions.
- You can define the model hierarchy without platform constraints.

### Single include path

```
### Compiling mdlref_top.c
    cl -Ireferenced_model_includes /Od /Oy- -DUSE_RTMODEL
-DMODEL=mdlref_top -DRT -DNUMST=2 -DTIDO1EQ=0 -DNCSTATES=0
-DMT=0 -DHAVESTDIO -c -DCRTAPI1=_cdecl -DCRTAPI2=_cdecl
-nologo -GS -D_X86_=1 -DWIN32 -D_WIN32 -W3 -D_WINNT
-D_WIN32_WINNT=0x0400 -D_WIN32_IE=0x0400 -DWINVER=0x0400
-D_MT -MT /wd4996 /fp:precise mdlref_top.c
mdlref_top.c
```

```
### Linking ...
    S:\Aslrtw\matlab\sys\perl\win32\bin\perl
S:\Aslrtw\matlab\rtw\c\tools\mkvc_lnk.pl mdlref_top.lnk
mdlref_top.obj mdlref_top_capi.obj mdlref_top_data.obj
rt_logging.obj rt_logging_nmi.obj rtw_modelmap_utils.obj
grt_main.obj rt_sim.obj
    link /RELEASE /INCREMENTAL:NO /NOLOGO
-subsystem:console,4.0 kernel32.lib ws2_32.lib mswsock.lib
advapi32.lib bufferoverflowu.lib libcpmt.lib
..\slprj\grt\sharedutils\rtwshared.lib @mdlref_top.lnk
@mdlref_top_ref.rsp -out:..\mdlref_top.exe
### Created executable mdlref_top.exe
```

### Response file for listing libraries

# Efficient Incremental Code Generation

## Challenge

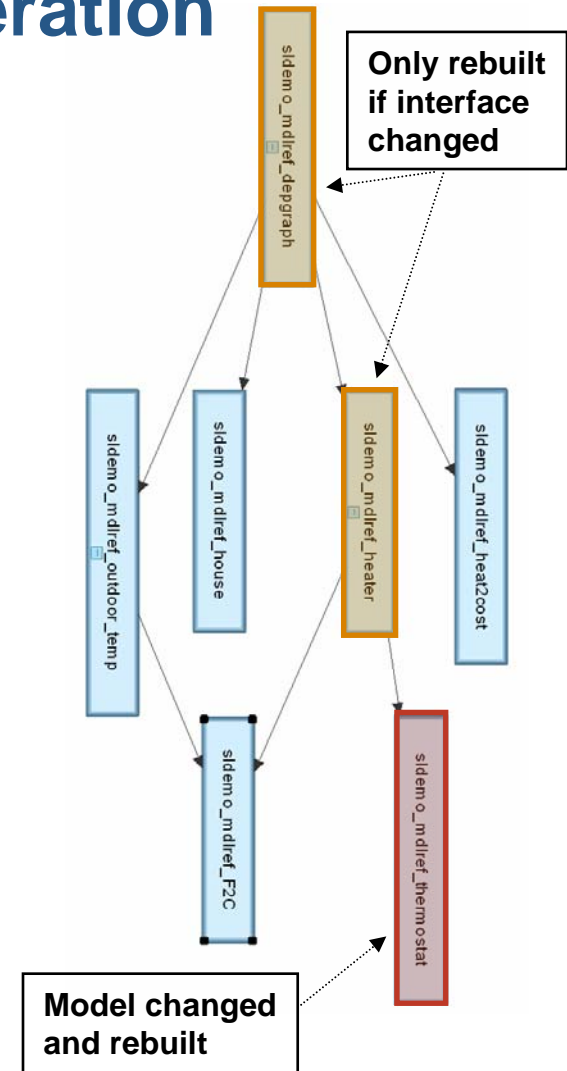
- Changing a referenced model caused all models that directly or indirectly reference it to rebuild, even if the code of the referencing models did not change.

## Solution

- An unedited model will rebuild only if the interface of a model it references changes.

## Benefit

- Incremental code generation works in more situations.
- Rebuilding after changing referenced models is faster.



```
>> rtwdemo_mdleftop
>> sldemo_mdiref_depgraph
```

# Agenda

- Simulink
  - Simulink Library Browser Enhancements
  - Simulation Performance
  - Component-Based Modeling
  - **Embedded MATLAB**
  - Data Management
  - File Management
  - Block Enhancements
  - User Interface Enhancements

# Support for Nontunable Structure Parameters

## Challenge

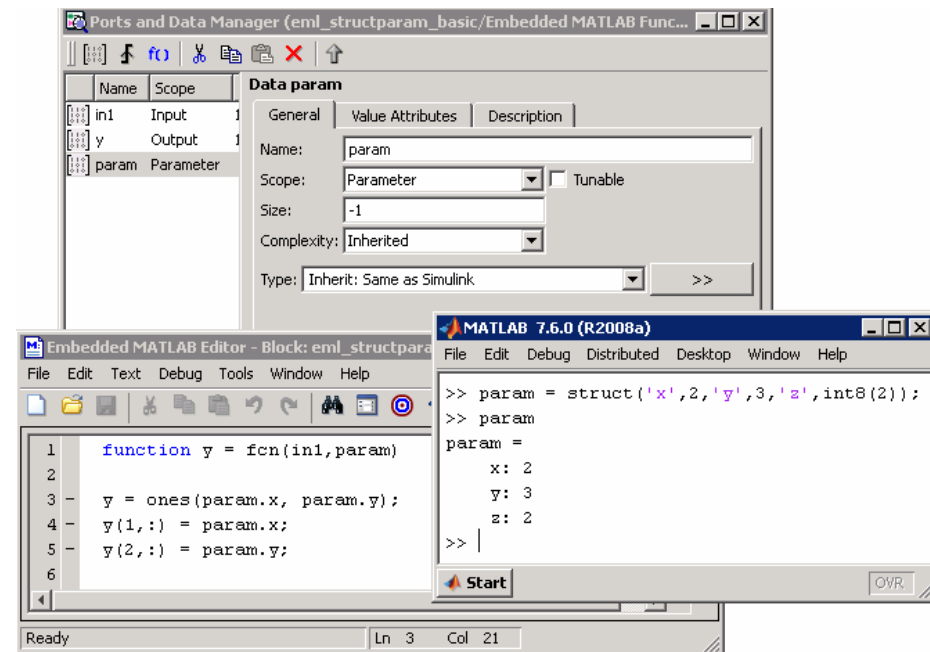
- It was not possible to bring structure parameters into the Embedded MATLAB Function block.

## Solution

- A structure can be defined and passed as a nontunable parameter to the block.

## Benefit

- Function line is smaller.
- Structure easily grows with the model.
- Interface with Simulink is stable.



# Support for Compiling P-Code

## Challenge

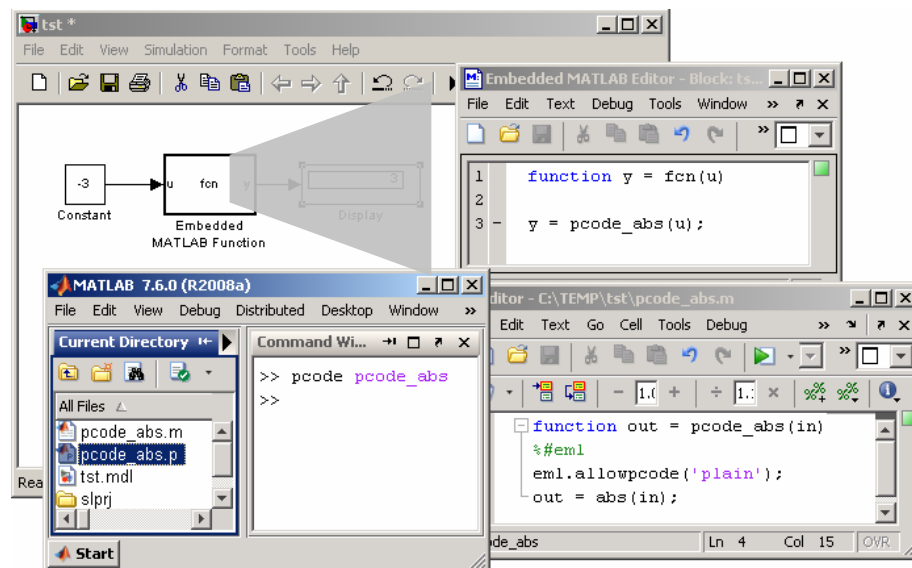
- Embedded MATLAB source code did not provide IP protection.

## Solution

- You can include `eml.allowpcode('plain')` in your M-file and then use `pcode` to compile your function.
- Your clients can now compile your P-coded function without needing the M-file.

## Benefit

- You can now protect your IP and support plain embeddable C code generation from your P-code.



# Support for Varargin/Varargout

## Challenge

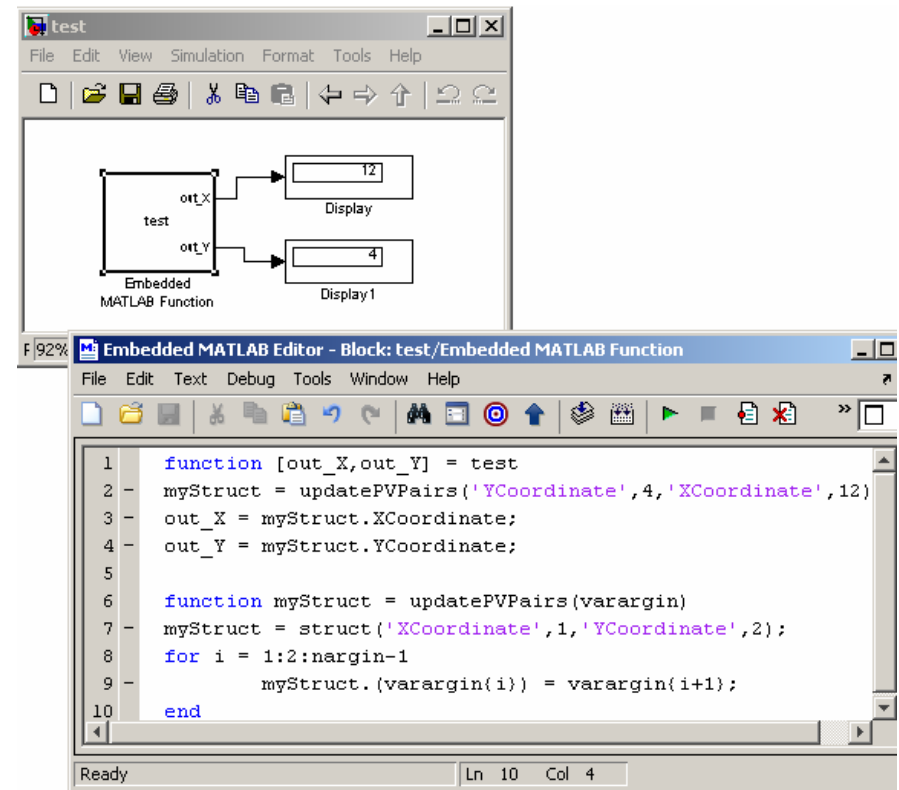
- There was a need to pass/return a variable number of inputs/outputs to an Embedded MATLAB function.

## Solution

- You can now use `varargin` and `varargout`.

## Benefit

- You can create Embedded MATLAB functions that take a variable number of inputs and return a variable number of outputs.



# Support for Controlling Function Inlining

## Challenge

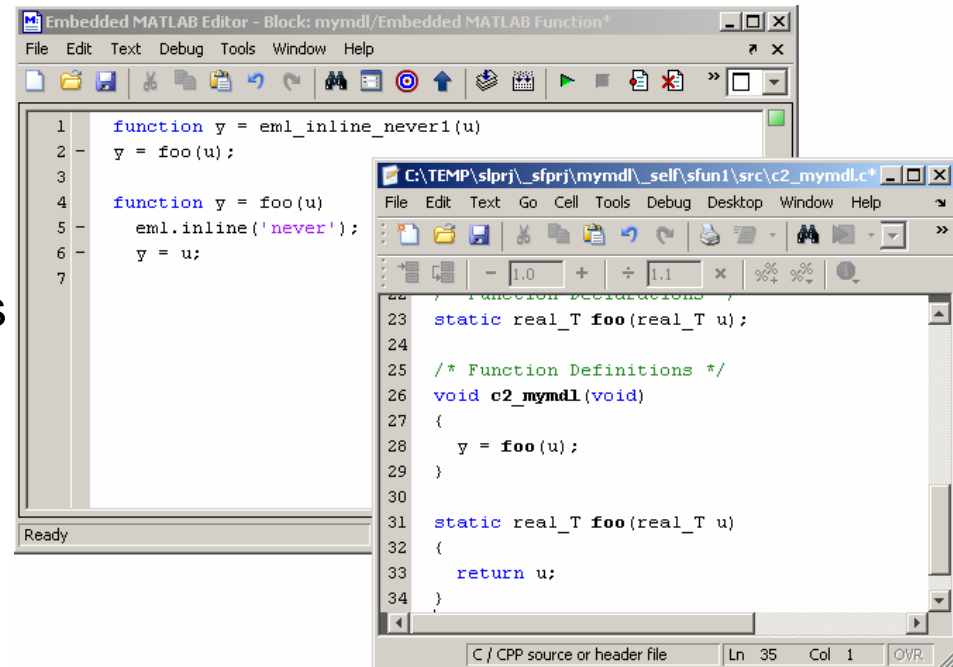
- There was a need to control inlining in the generated code.

## Solution

- `eml.inline('always')` forces inlining of the current function.
- `eml.inline('never')` prevents inlining of the current function.

## Benefit

- You can control the appearance and traceability of the generated code.



The top screenshot shows the Embedded MATLAB Editor with the following code:

```

1  function y = eml_inline_never1(u)
2  - y = foo(u);
3
4  function y = foo(u)
5  -     eml.inline('never');
6  -     y = u;
7
    
```

The bottom screenshot shows the generated C code in a text editor:

```

23  static real_T foo(real_T u);
24
25  /* Function Definitions */
26  void c2_mymdl(void)
27  {
28      y = foo(u);
29  }
30
31  static real_T foo(real_T u)
32  {
33      return u;
34  }
    
```

# EMLC and EMLMEX Support Constant Inputs

## Challenge

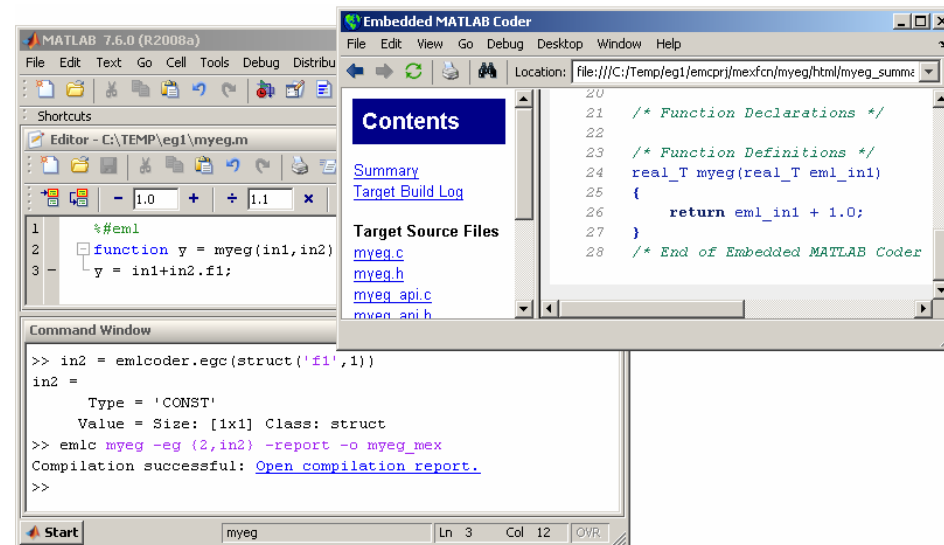
- Function can be parameterized by a value that doesn't change at run time. The generated code incurs the overhead associated with this value when it's really a constant.

## Solution

- You can use `emlcoder.egc` to specify a constant input.

## Benefit

- Generated code can run faster.
- Embedded MATLAB will permit this value to be used where constants are required.



The screenshot displays the MATLAB 7.6.0 (R2008a) environment. The main Editor window shows a script named `myeg.m` with the following code:

```

1 %#eml
2 function y = myeg(in1,in2)
3     y = in1+in2.f1;
  
```

The Command Window shows the following commands and output:

```

>> in2 = emlcoder.egc(struct('f1',1))
in2 =
    Type = 'CONST'
    Value = Size: [1x1] Class: struct
>> emlc myeg -eg {2,in2} -report -o myeg_mex
Compilation successful: Open compilation report.
>>
  
```

The Embedded MATLAB Coder window shows the generated C code:

```

20
21 /* Function Declarations */
22
23 /* Function Definitions */
24 real_T myeg(real_T eml_in1)
25 {
26     return eml_in1 + 1.0;
27 }
28 /* End of Embedded MATLAB Coder
  
```

# EMLC and EMLMEX Support More Control over Compiler Options

## Challenge

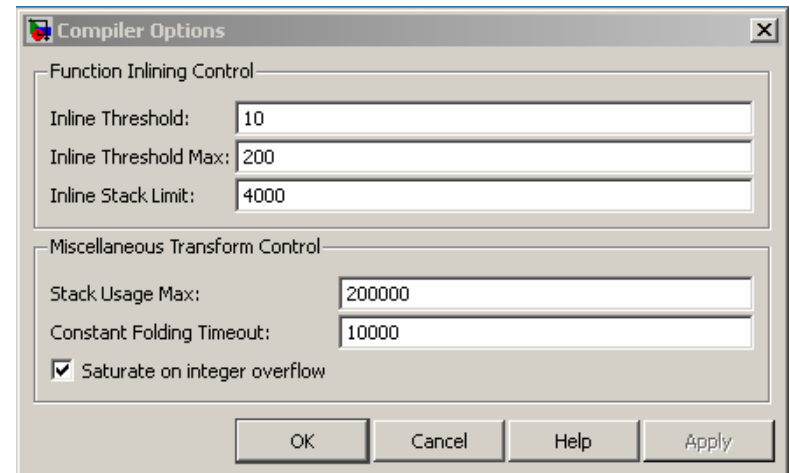
- There was a need for more control over how `emlc` and `emlmex` operate.

## Solution

- Use `emlcoder.CompilerOptions` to control:
  - Inlining threshold
  - Stack usage
  - Integer saturation code

## Benefit

- You have more control over generated code.



# Agenda

- Simulink
  - Simulink Library Browser Enhancements
  - Simulation Performance
  - Component-Based Modeling
  - Embedded MATLAB
  - **Data Management**
  - File Management
  - Block Enhancements
  - User Interface Enhancements

# Fixed-Point Details Displayed

## Challenge

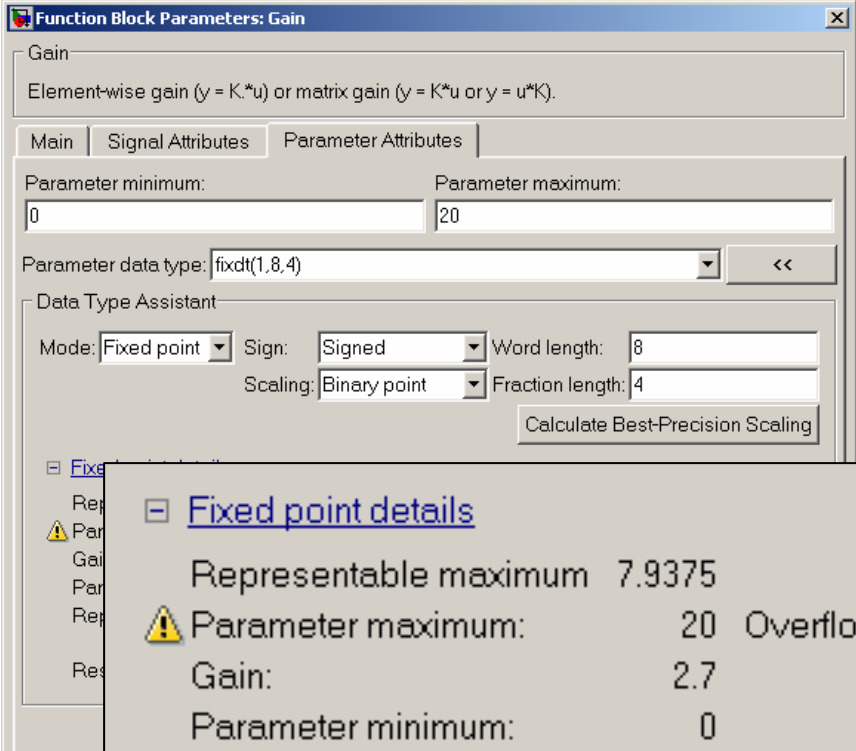
- Understanding the implications of fixed-point data types can be difficult.

## Solution

- Block's parameter dialog box now shows the following fixed-point details:
  - Resolution (precision)
  - Representable min./max.
  - Associated numeric parameters

## Benefit

- It is easier to understand the implications when specifying fixed-point data types.



The screenshot shows the 'Function Block Parameters: Gain' dialog box. The 'Parameter Attributes' tab is active, displaying the following settings:

- Parameter minimum: 0
- Parameter maximum: 20
- Parameter data type: `fixdt(1,8,4)`
- Data Type Assistant:
  - Mode: Fixed point
  - Sign: Signed
  - Word length: 8
  - Scaling: Binary point
  - Fraction length: 4
  - Calculate Best-Precision Scaling button

An expanded 'Fixed point details' window is overlaid on the dialog, showing the following information:

| Fixed point details   |             |
|-----------------------|-------------|
| Representable maximum | 7.9375      |
| ⚠ Parameter maximum:  | 20 Overflow |
| Gain:                 | 2.7         |
| Parameter minimum:    | 0           |
| Representable minimum | -8          |
| Resolution            | 0.0625      |

# Enhanced Bus Editor

## Challenge

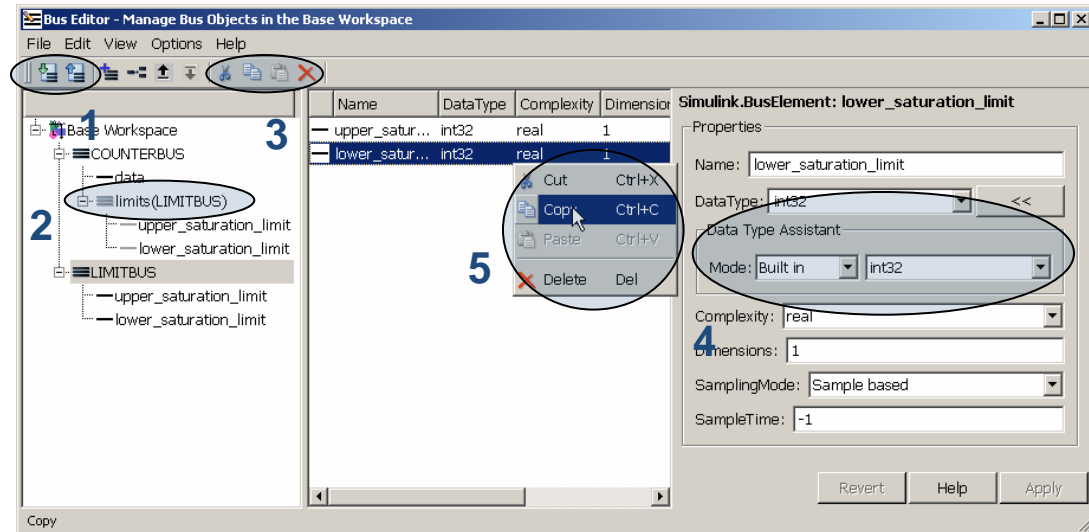
- Defining and managing bus objects can be difficult.

## Solution

- Bus Editor has been enhanced.

## Benefit

- You can now:
  - Import/export bus objects from MAT-files and M-files
  - Explore bus hierarchy
  - Cut, Copy, Paste, Delete
  - Define bus elements with Data Type Assistant
  - Use context menus



```
>> sldemo_mdlref_bus
```

# Signal Resolution Icons in the Simulink Editor

## Challenge

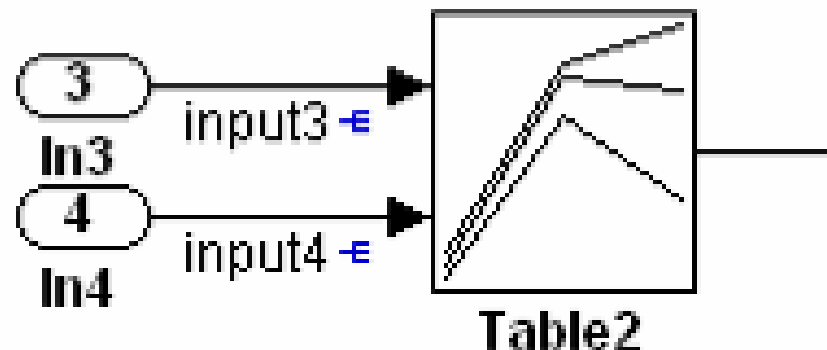
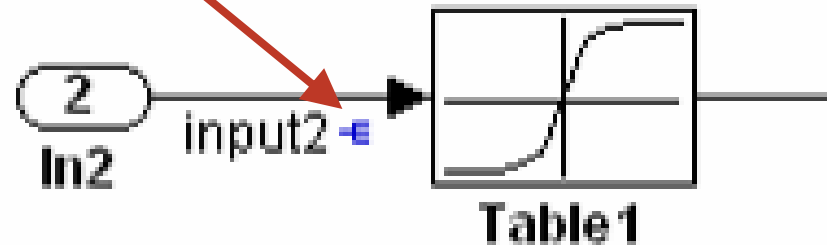
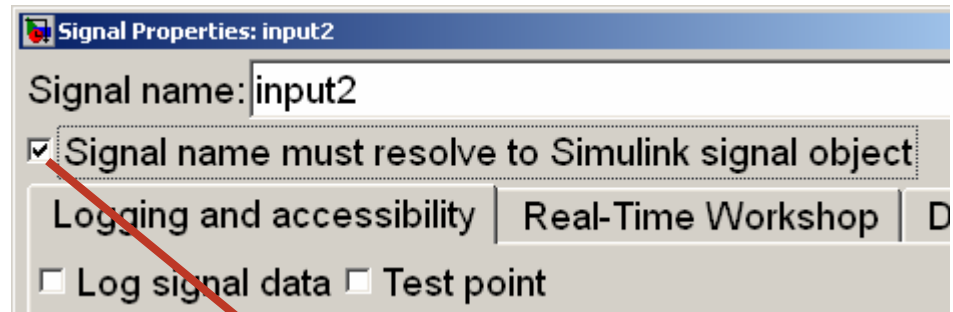
- Determining whether a signal label is intended to resolve to a signal object can be difficult.

## Solution

- You can add icons to signals marked as “Signal name must resolve to Simulink signal object.”

## Benefit

- It is easier see which signal labels are intended to resolve to signal objects when viewing or printing models that use Simulink signal objects.



# Data Logging in 64-Bit Platforms

## Challenge

- Logging of simulation data on 64-bit platforms was limited to 2 GB.

## Solution

- Simulink data logging on 64-bit platforms can now save more simulation data.
- When you log data using:
  - Structure, StructureWithTime, or Timeseries format, you can save  $2^{(48-1)}$  bytes in each field.
  - Array format, you can save  $2^{(48-1)}$  bytes in each array.

## Benefit

- You can save more simulation data from long simulations.

```
% Logging output signal using Array Format
```

```
>> whos
```

| Name | Size         | Bytes      | Class  |
|------|--------------|------------|--------|
| yout | 1024x2097152 | 8589934592 | single |

8 GB

```
% Logging output signal using Structure Format
```

```
>> yout.signals
```

```
ans =
```

```
values: [1024x2097152 single]
```

```
dimensions: 2097152
```

```
label: "
```

```
blockName: 'mmx64_logging_struct/Out1'
```

# Agenda

- Simulink
  - Simulink Library Browser Enhancements
  - Simulation Performance
  - Component-Based Modeling
  - Embedded MATLAB
  - Data Management
  - **File Management**
  - Block Enhancements
  - User Interface Enhancements

# Autosave Preference

## Challenge

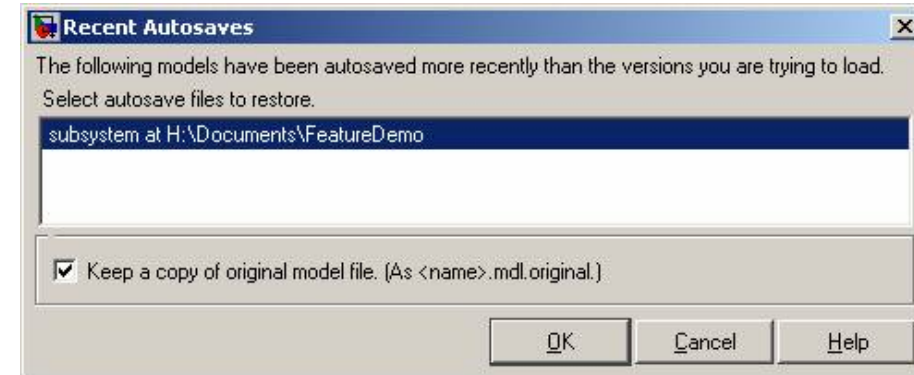
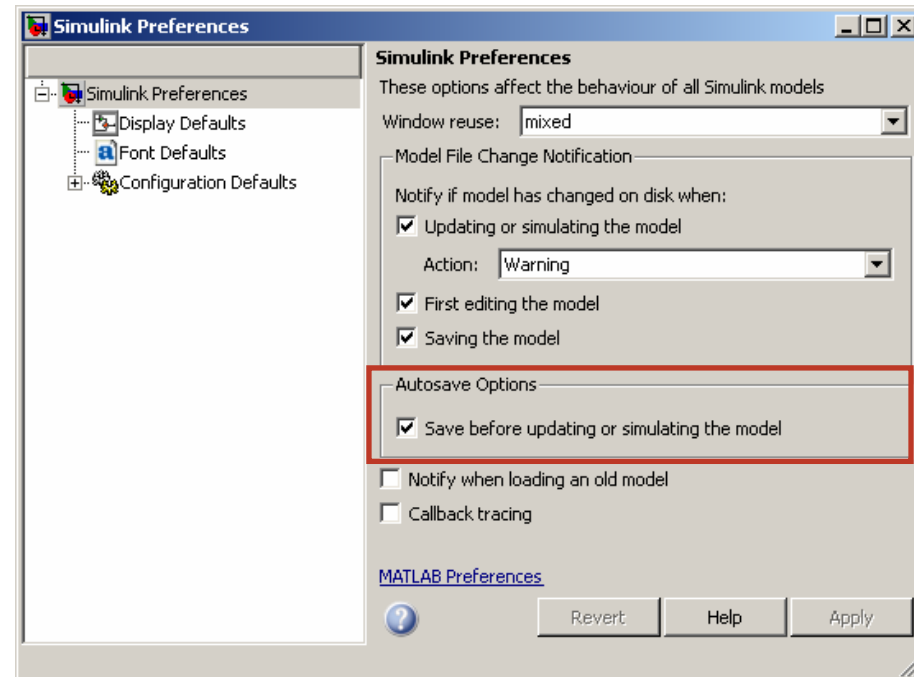
- Work was lost when models were closed without saving.

## Solution

- You can now set a preference to enable autosave.
- When a model is simulated, an autosave file is created.
- When a model is loaded, if a more recent autosave exists:
  - You'll see a warning.
  - You can restore the autosaved model.
  - You can keep a copy of the original.

## Benefit

- You can prevent the loss of your work.



# Notification When Loading an Old Model

## Challenge

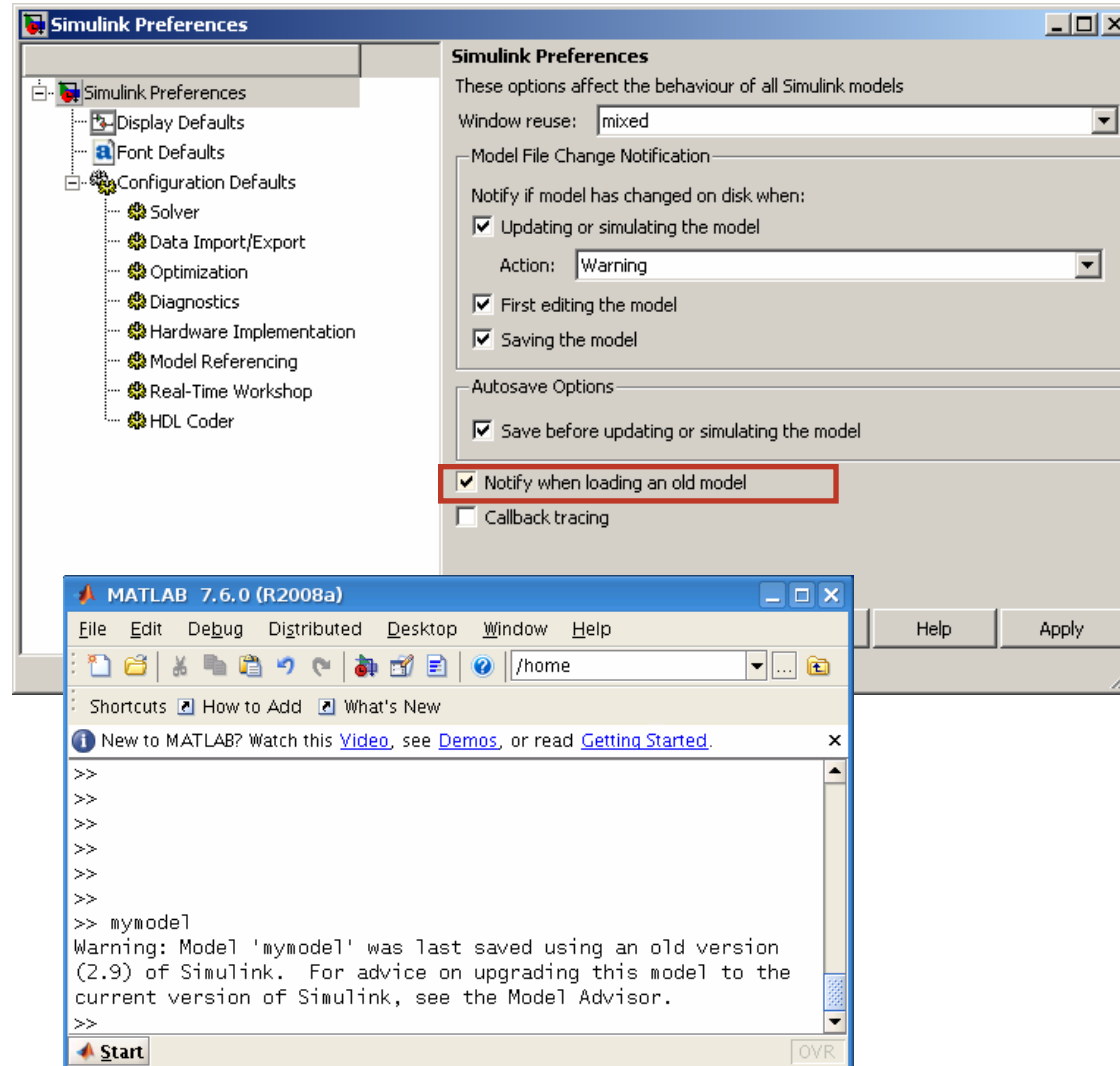
- Models that had been saved in older versions of Simulink may not be compatible with the latest release.
- Simulink did not indicate incompatibility when you loaded the model.

## Solution

- You can now set a Simulink preference to be notified when loading a model saved in an older version of Simulink.

## Benefit

- You are reminded that you can take advantage of Model Advisor checks and `slupdate` feature to update models.



The image shows two overlapping windows from the MATLAB environment. The top window is the 'Simulink Preferences' dialog box. In the 'Model File Change Notification' section, the 'Notify when loading an old model' checkbox is checked and highlighted with a red rectangle. Below it, the 'Callback tracing' checkbox is unchecked. The bottom window is the MATLAB command window for version 7.6.0 (R2008a). It shows a warning message: 'Warning: Model 'myModel' was last saved using an old version (2.9) of Simulink. For advice on upgrading this model to the current version of Simulink, see the Model Advisor.'

# Simulink Manifest Tools

## Challenge

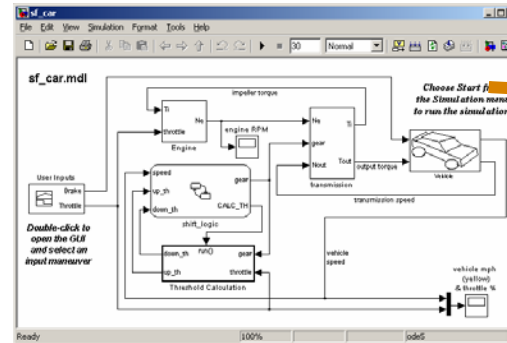
- Some common file dependencies were not detected by Simulink Manifest Tools.
- Important files needed for the model could be missing when a model was exported.

## Solution

- Simulink Manifest Tools now has the ability to:
  - Detect TLC files required by S-functions
  - Detect common data loading functions (`xlsread`, `csvread`, `textread`, `importdata`, `dlmread`)
  - Detect `.fig` files created by GUIDE

## Benefit

- A larger set of dependent files are automatically detected, reducing the number of files you need to add manually to the manifest.



Analyze

Export to zip file

| File Name  | Size        | Last Modified Date  | Exportable |
|--|-------------|---------------------|------------|
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain.mdl      | 6274 bytes  | 2006-10-27 17:11:56 | Exportable |
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain_data.mat | 1976 bytes  | 2006-10-27 17:12:03 |            |
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain0.mdl     | 34759 bytes | 2006-09-08 14:13:14 |            |
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain1.mdl     | 19090 bytes | 2006-09-08 14:13:11 |            |
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain2.mdl     | 29629 bytes | 2006-09-08 14:13:15 |            |
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain3.mdl     | 17 bytes    | 2006-10-27 17:11:59 |            |
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain4.mdl     | 30 bytes    | 2006-10-27 17:12:00 |            |
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain5.mdl     | 4 bytes     | 2006-10-27 17:12:01 |            |
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain6.mdl     | 11642 bytes | 2006-10-27 17:11:57 |            |
| C:\Users\j_burke\Documents\Simulink\manifest_analysis\mpowertrain7.mdl     | 20390 bytes | 2006-10-27 17:11:58 |            |

Report

# Agenda

- Simulink
  - Simulink Library Browser Enhancements
  - Simulation Performance
  - Component-Based Modeling
  - Embedded MATLAB
  - Data Management
  - File Management
  - **Block Enhancements**
  - User Interface Enhancements

# New Discrete FIR Filter Block

## Challenge

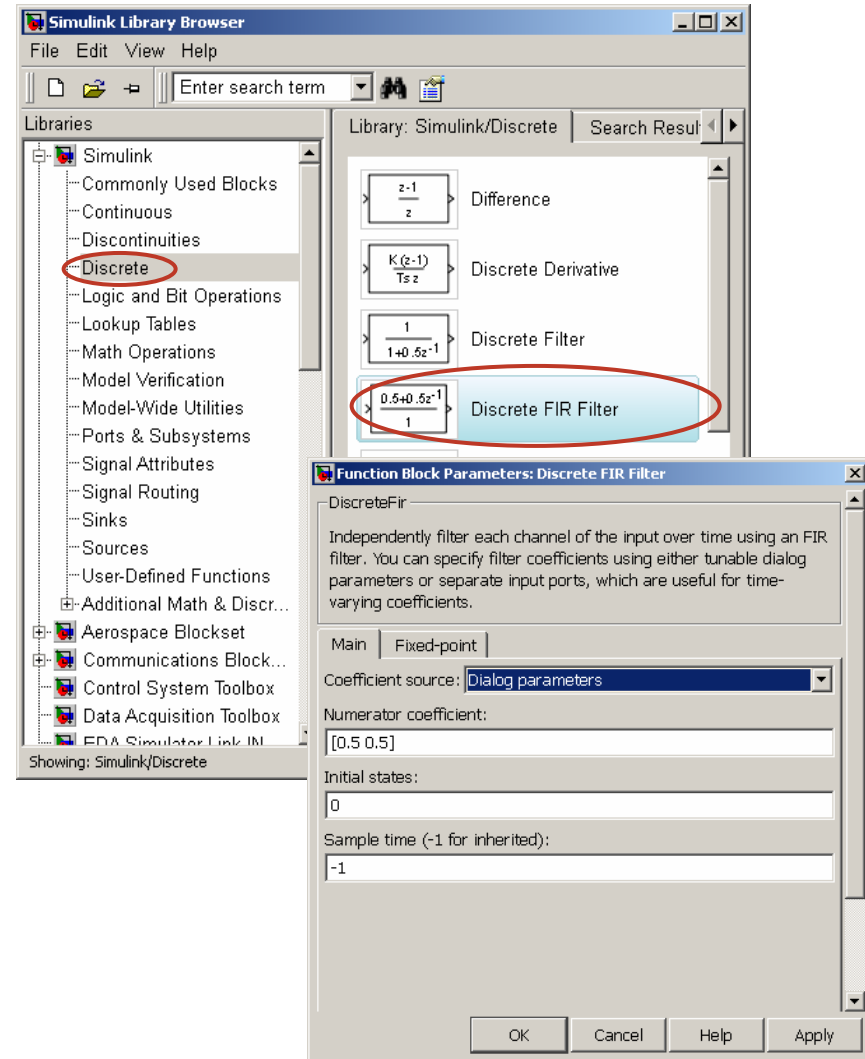
- Weighted Moving Average filter block did not support multiple channel filtering.

## Solution

- The new Discrete FIR Filter block replaces the Weighted Moving Average block from Simulink and independently filters multichannel signals.
- You can use `slupdate` to update existing models that use Weighted Moving Average blocks.

## Benefit

- Reduce the number of filter blocks in the model
- Eliminate the need to separate multichannel signals to apply a FIR filter
- Increase readability of the design



# Enhanced Rate Transition Block Specification

## Challenge

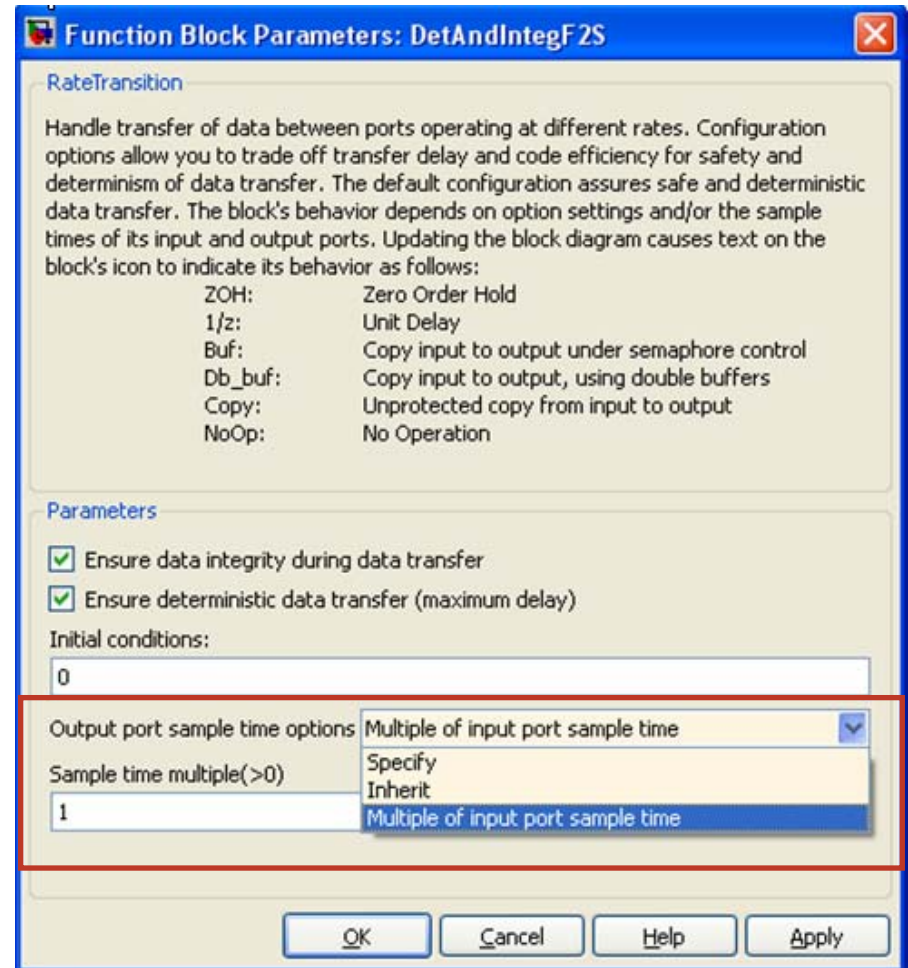
- The Rate Transition block output port sample rate could be specified explicitly or inherited from the model.
- This made it difficult to use in Simulink libraries and other scenarios where the ratio, but not the values themselves, needed to be “locked down.”

## Solution

- A new dialog box option specifies output port sample time as a multiple of input port sample time.

## Benefit

- Usability for multirate/multitasking modeling is improved.



# Improved Rate Transition Block Insertion

## Challenge

- Auto insertion of Rate Transition blocks was configured conservatively for deterministic data transfer.
- This is too conservative for some applications.

## Solution

- A new option in the Configuration Parameters dialog box enables you to control determinism.

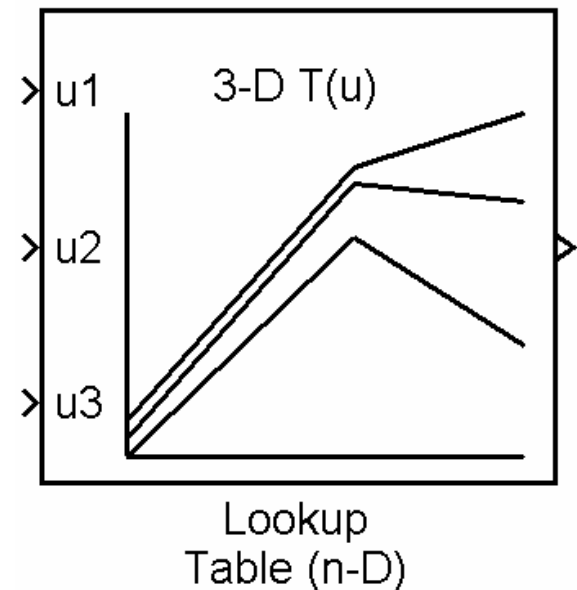
## Benefit

- Usability is improved for multirate/multitasking modeling where auto insertion of Rate Transition blocks is desired.

|  |                                 |
|--|---------------------------------|
| Simulation time  |                                 |
| Start time:  | 0.0                             |
| Stop time:   | 0.5                             |
| Solver options   |                                 |
| Type:  | Fixed-step                      |
| Solver:  | discrete (no continuous states) |
| Fixed-step size (fundamental sample time):   | 1/2000                          |
| Tasking and sample time options  |                                 |
| Periodic sample time constraint:   | Unconstrained                   |
| Tasking mode for periodic sample times:  | MultiTasking                    |
| <input checked="" type="checkbox"/> Automatically handle rate transition for data transfer |                                 |
| Deterministic data transfer:   | Whenever possible               |
|  | Whenever possible               |
|  | Always                          |
|  | Never (minimum delay)           |

# Lookup Table (n-D) Block

- Lookup Table (n-D) now supports:
  - All numeric types, including fixed point (except for linear extrapolation and cubic spline interpolation)
  - Vector signals (except for cubic spline interpolation)
  - Complex table data (except for cubic spline interpolation)
- The block is MISRA-C compliant; there is no pointer arithmetic in generated utility functions.
- The Fixed-Point Function Approximation demo has been enhanced to use this block.



# Accumulator Data Type

## Challenge

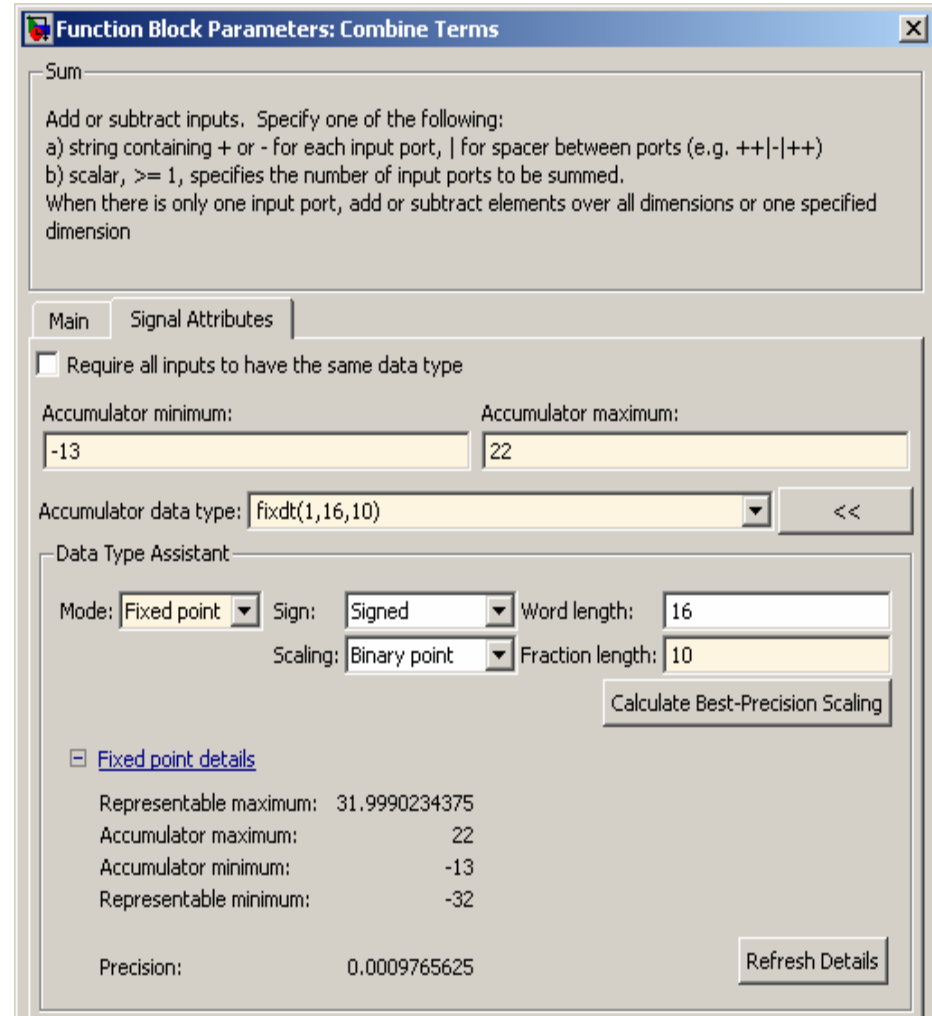
- Sum block did not provide data type control over its internal accumulator.
- This prevented designers from making tradeoffs between accuracy and efficiency.

## Solution

- Accumulator data type is exposed as a parameter to the Sum block.

## Benefit

- You can control design tradeoffs.
- You have greater control of the generated code.



# Agenda

- Simulink
  - Simulink Library Browser Enhancements
  - Simulation Performance
  - Component-Based Modeling
  - Embedded MATLAB
  - Data Management
  - File Management
  - Block Enhancements
  - **User Interface Enhancements**

# Simulink Preferences

## Challenges

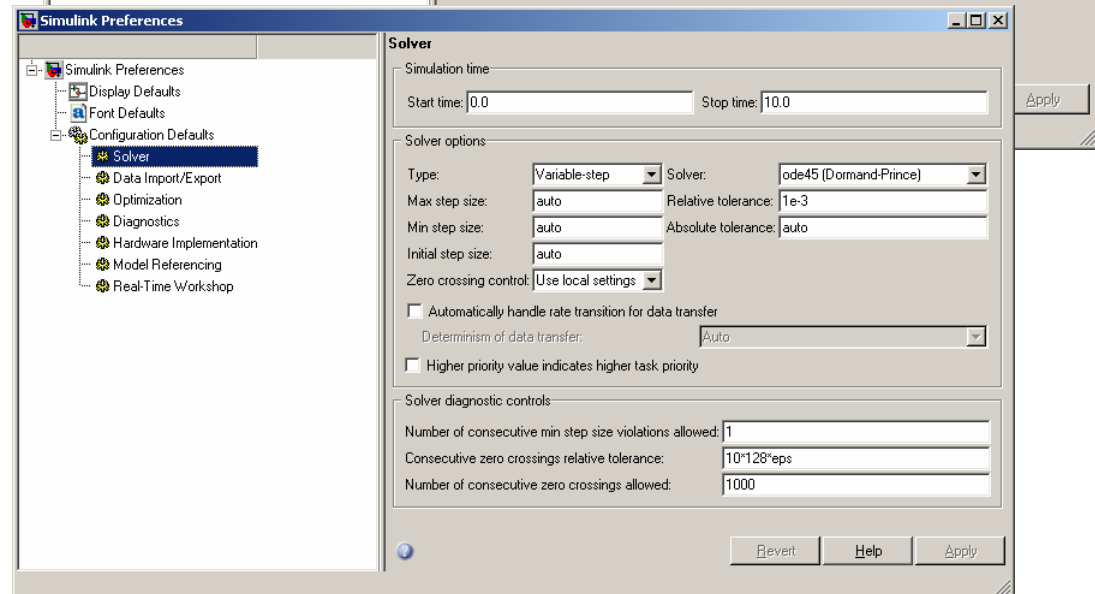
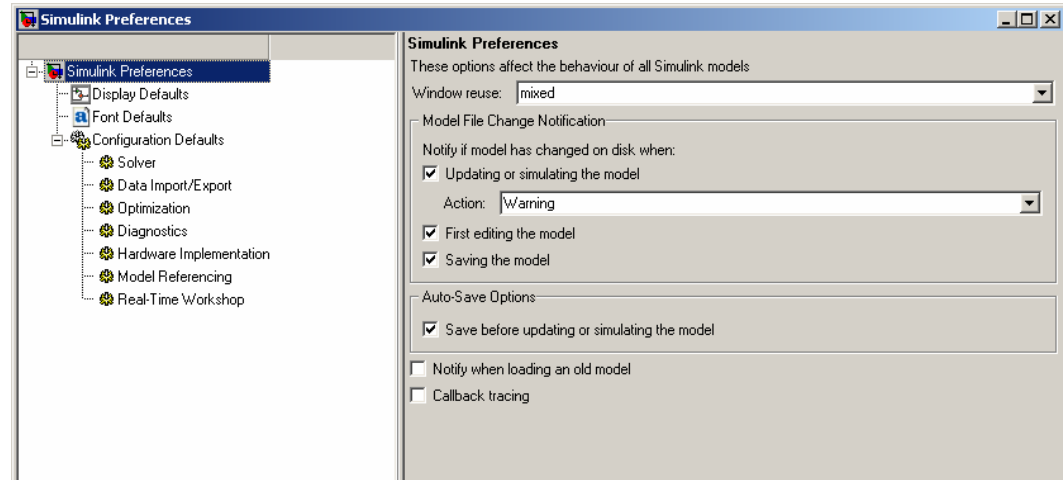
- Simulink Preferences were split between two locations:
  - Model Explorer
  - MATLAB Preferences

## Solution

- A new window gives you a single point of access to all Simulink Preferences.

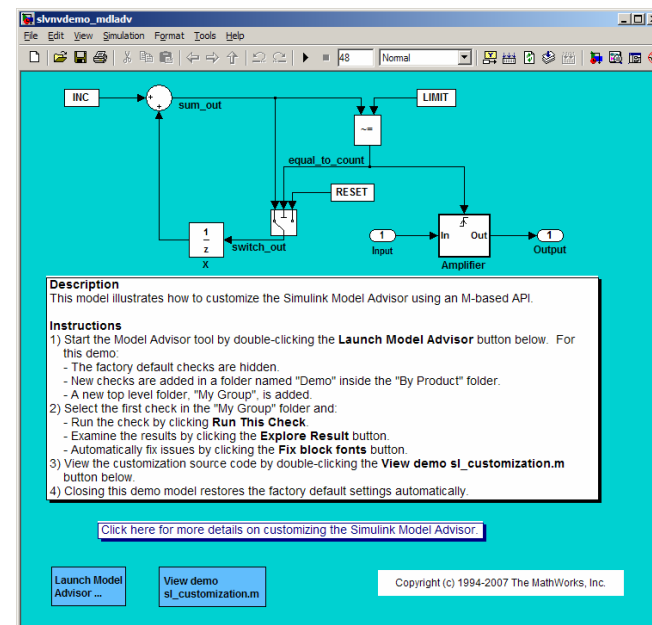
## Benefit

- It is easier to find and change environment settings and defaults for new block diagrams.



# Model Advisor Infrastructure Highlights

- Reset context menu option
  - Clears folder results so you can restart
- Enhanced report format
  - All checks are contained in report
  - Added time stamps
- Explore Result button
  - Mass/batch edits to all objects at fault
- Significantly enhanced API
  - Control entire Model Advisor tree
  - Check input parameters
- Updated demo *slvndemo\_mdldv*

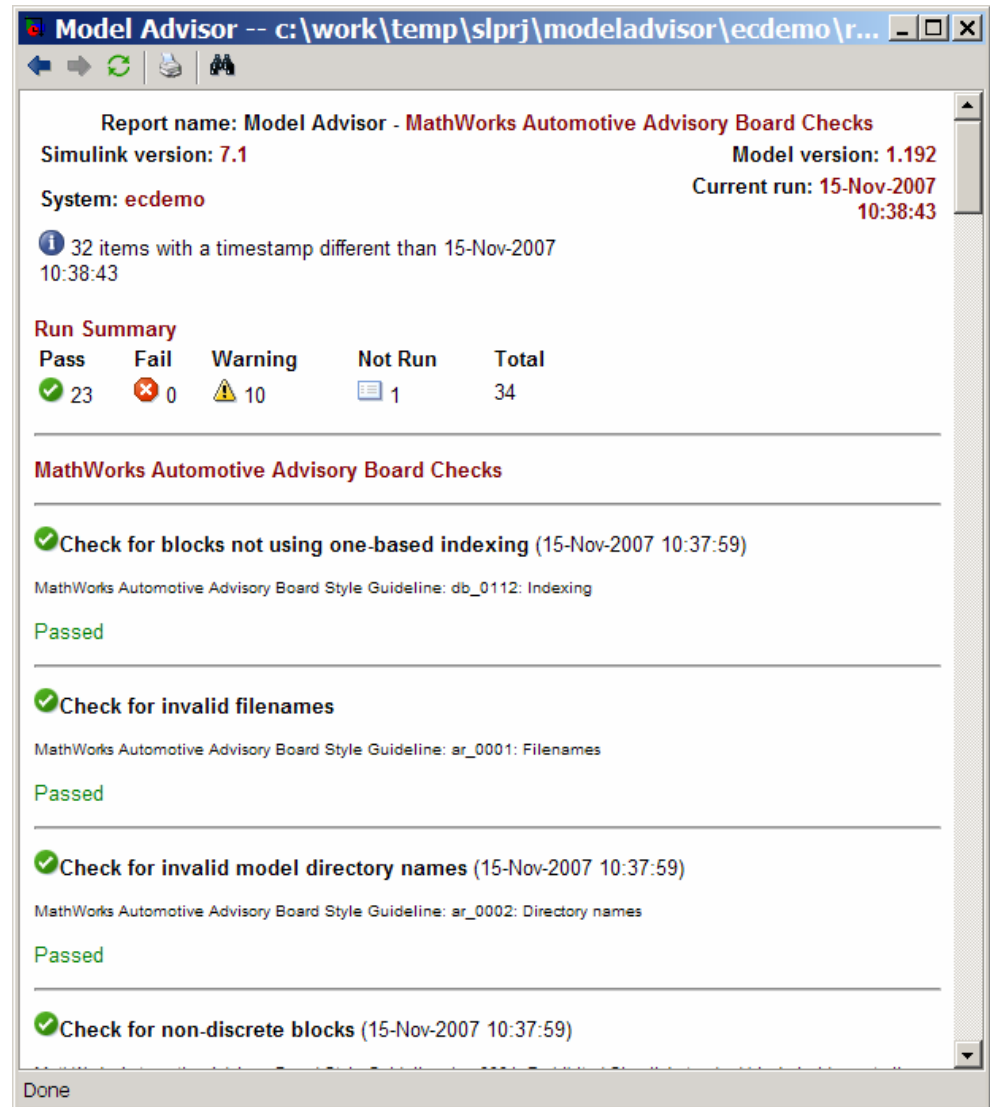


```
>>slvndemo_mdldv
```

# Enhanced Model Advisor Report

This report is now more useful as a process audit document:

- Provides detailed summary
- Indicates valid check states: Pass, Fail, Warning, and Not Run
- Follows exact order of the Model Advisor tree
- Generates reports for any folder on the fly
- Documents when checks are run with individualized time stamps



Model Advisor -- c:\work\temp\slprj\modeladvisor\ecdemo\r...

Report name: Model Advisor - MathWorks Automotive Advisory Board Checks  
 Simulink version: 7.1  
 System: ecdemo  
 Model version: 1.192  
 Current run: 15-Nov-2007 10:38:43

32 items with a timestamp different than 15-Nov-2007 10:38:43

| Run Summary |      |         |         |       |
|-------------|------|---------|---------|-------|
| Pass        | Fail | Warning | Not Run | Total |
| 23          | 0    | 10      | 1       | 34    |

**MathWorks Automotive Advisory Board Checks**

- ✓ Check for blocks not using one-based indexing (15-Nov-2007 10:37:59)  
 MathWorks Automotive Advisory Board Style Guideline: db\_0112: Indexing  
 Passed
- ✓ Check for invalid filenames  
 MathWorks Automotive Advisory Board Style Guideline: ar\_0001: Filenames  
 Passed
- ✓ Check for invalid model directory names (15-Nov-2007 10:37:59)  
 MathWorks Automotive Advisory Board Style Guideline: ar\_0002: Directory names  
 Passed
- ✓ Check for non-discrete blocks (15-Nov-2007 10:37:59)

Done

# Enhanced Controls for Solver Pane

R2007b

## Challenge

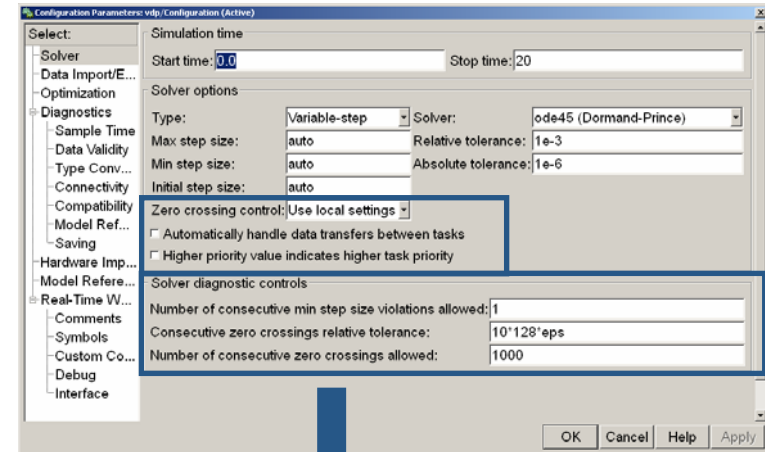
- Solver pane in the Configuration Parameters UI had related options in both Solver options and Solver diagnostic controls panes.
- Existing layout was not sufficient to support new zero crossing options.

## Solution

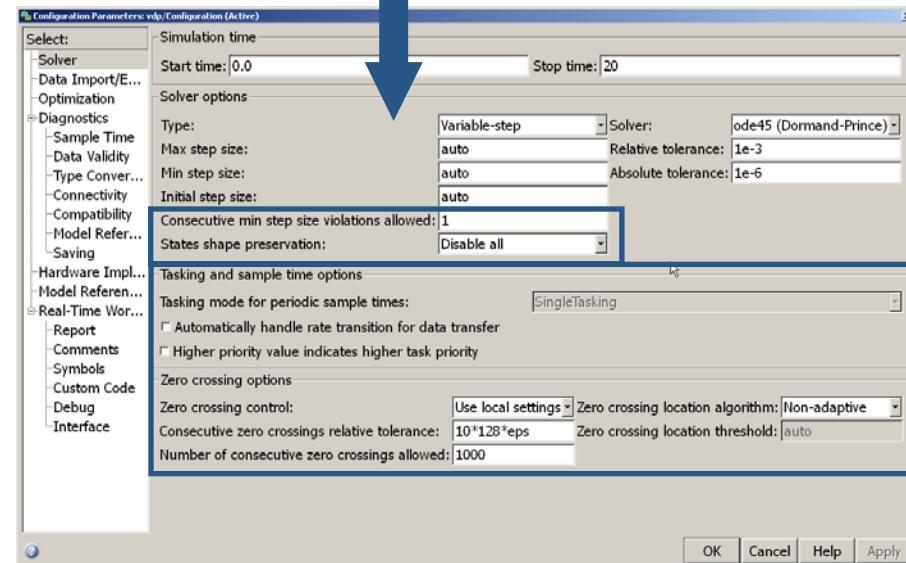
- Options are reorganized into like groupings and new panes have been created:
  - Tasking and sample time options
  - Zero crossing options

## Benefit

- Setting up solver settings is easier.



R2008a



# Context-Sensitive Help

## Challenge

- It can be difficult to find help on all the options in the Configuration Parameters and Preferences dialog boxes when you are setting those options.

## Solution

- You can now right-click the option label and select “What’s This?”

## Benefit

- You have immediate access to help for any of the hundreds of options on the Configuration Parameters dialog box or the Simulink Preferences dialog box.
- The learning curve for using Simulink is reduced.

