

SimEvents® Release Notes

How to Contact The MathWorks



www.mathworks.com Web
comp.soft-sys.matlab Newsgroup
www.mathworks.com/contact_TS.html Technical Support



suggest@mathworks.com Product enhancement suggestions
bugs@mathworks.com Bug reports
doc@mathworks.com Documentation error reports
service@mathworks.com Order status, license renewals, passcodes
info@mathworks.com Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



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

For contact information about worldwide offices, see the MathWorks Web site.

SimEvents[®] Release Notes

© COPYRIGHT 2005–2010 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Summary by Version	1
Version 3.1 (R2010a) SimEvents Software	4
Version 3.0 (R2009b) SimEvents Software	11
Version 2.4 (R2009a) SimEvents Software	13
Version 2.3 (R2008b) SimEvents Software	18
Version 2.2 (R2008a) SimEvents Software	20
Version 2.1 (R2007b) SimEvents Software	23
Version 2.0 (R2007a) SimEvents Software	34
Version 1.2 (R2006b) SimEvents Software	43
Version 1.1 (R2006a) SimEvents Software	51
Version 1.0 (R14SP3+) SimEvents Software	53
Compatibility Summary for SimEvents Software	55

Summary by Version

This table provides quick access to what's new in each version. For clarification, see "Using Release Notes" on page 1.

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Latest Version V3.1 (R2010a)	Yes Details	Yes Summary	Bug Reports Includes fixes	Printable Release Notes: PDF Current product documentation
V3.0 (R2009b)	Yes Details	No	Bug Reports Includes fixes	No
V2.4 (R2009a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V2.3 (R2008b)	Yes Details	No	Bug Reports Includes fixes	No
V2.2 (R2008a)	Yes Details	No	Bug Reports Includes fixes	No
V2.1 (R2007b)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V2.0 (R2007a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V1.2 (R2006b)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V1.1 (R2006a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V1.0 (R14SP3+)	Yes Details	Not applicable	Bug Reports	No

Using Release Notes

Use release notes when upgrading to a newer version to learn about:

- New features
- Changes
- Potential impact on your existing files and practices

Review the release notes for other MathWorks™ products required for this product (for example, MATLAB® or Simulink®). Determine if enhancements, bugs, or compatibility considerations in other products impact you.

If you are upgrading from a software version other than the most recent one, review the current release notes and all interim versions. For example, when you upgrade from V1.0 to V1.2, review the release notes for V1.1 and V1.2.

What Is in the Release Notes

New Features and Changes

- New functionality
- Changes to existing functionality

Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility between versions, the **Compatibility Considerations** subsection explains the impact.

Compatibility issues reported after the product release appear under Bug Reports at The MathWorks™ Web site. Bug fixes can sometimes result in incompatibilities, so review the fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

The MathWorks offers a user-searchable Bug Reports database so you can view Bug Reports. The development team updates this database at release time and as more information becomes available. Bug Reports include provisions for any known workarounds or file replacements. Information is available for bugs existing in or fixed in Release 14SP2 or later. Information is not available for all bugs in earlier releases.

Access Bug Reports using your MathWorks Account.

Version 3.1 (R2010a) SimEvents Software

This table summarizes what's new in Version 3.1 (R2010a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes Summary	Bug Reports Includes fixes	Printable Release Notes: PDF Current product documentation

New features and changes introduced in this version are

- “Block-Based Breakpoints in Debugger” on page 4
- “Block Operations Information in Debugger” on page 5
- “Changes in Behavior of Pending Entity Signals” on page 5
- “Renaming of Parameter to Enable Pending Entity Signal” on page 9
- “Expanded Options for Opening Release Gate” on page 9
- “Blocks in Attributes Library Must Get or Set at Least One Attribute” on page 10
- “Parameters and Parameter Values Being Removed” on page 10

Block-Based Breakpoints in Debugger

With the SimEvents® debugger, you can investigate the behavior of particular blocks using block-based breakpoints. After you establish a breakpoint on a block, the debugger suspends the simulation when that block is about to perform certain operations. For details, see these resources:

- `sedb.blkbreak` function reference page
- “Defining a Breakpoint”
- “Using Breakpoints During Debugging”
- “Block Operations Relevant for Block Breakpoints”

Block Operations Information in Debugger

These blocks now include their operations in the simulation log:

- Discrete Event Signal to Workspace
- Instantaneous Event Counting Scope
- Signal Scope
- X-Y Signal Scope

These blocks now appear in the output of the `sedb.blklist` function and are valid as inputs to the `sedb.blkinfo` function:

- Discrete Event Signal to Workspace
- Discrete Event Subsystem
- Instantaneous Event Counting Scope

The `sedb.blklist` function sorts its Command Window output and cell array output by block names instead of by block identifiers.

Compatibility Considerations

If you have code that manipulates or indexes into the output cell array from `sedb.blklist`, you might need to update the code to reflect new rows or a different sequence of rows.

Changes in Behavior of Pending Entity Signals

The blocks in the following table have an optional `pe` or `#pe` signal output port. The signals at these ports provide information about pending entities in the block. The port behaviors are now simpler and more consistent across the various blocks.

Block	Has Optional <code>pe</code> Port	Has Optional <code>#pe</code> Port
Event-Based Entity Generator	Yes	No
Infinite Server	Yes	Yes

Block	Has Optional pe Port	Has Optional #pe Port
N-Server	Yes	Yes
Output Switch	Yes	No
Single Server	Yes	No
Time-Based Entity Generator	Yes	No

In V3.1 (R2010a), a pending entity is an entity that has tried and failed to depart from the block in which the entity resides.

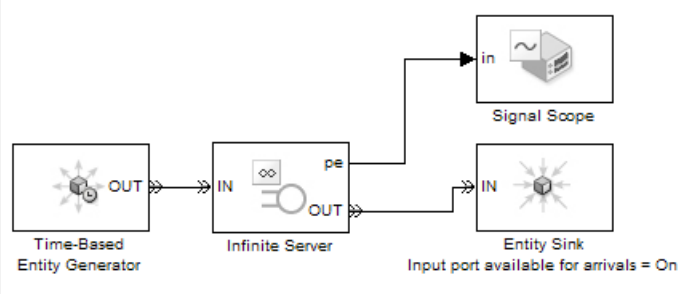
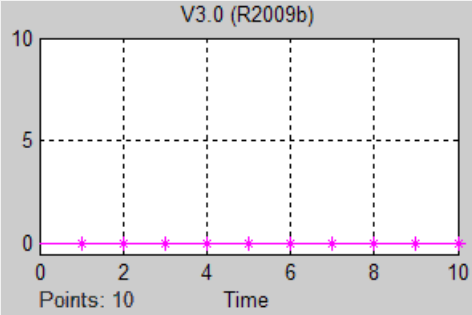
When a block produces a **pe** output signal, the signal has an update (that is, a sample time hit) whenever there is a change in the set of pending entities that the block stores. The signal value is:

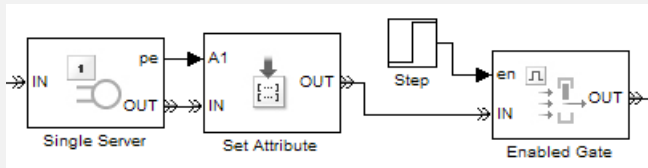
- 1, if the block stores one or more pending entities
- 0, if the block does not store any pending entities

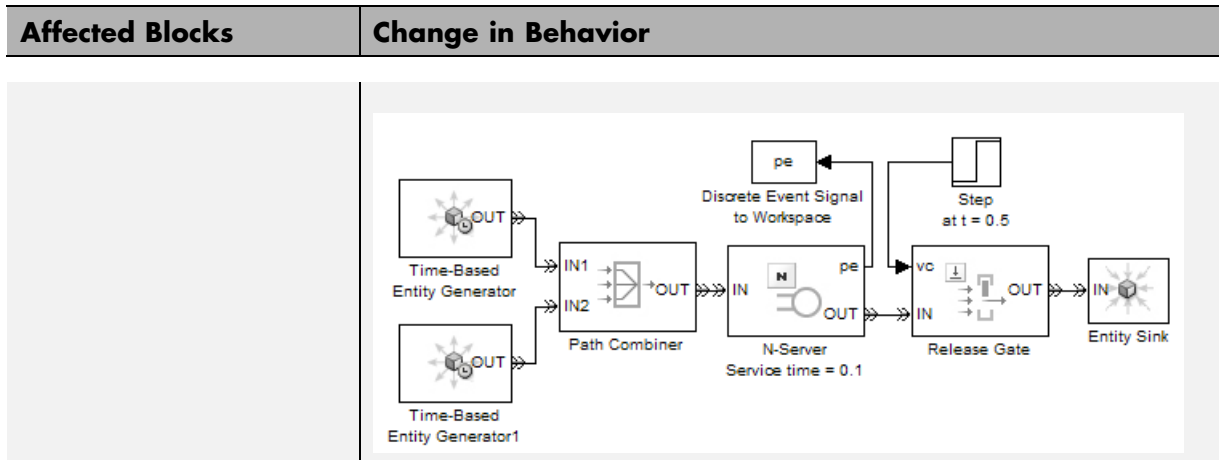
When a block produces a **#pe** output signal, the signal has an update whenever there is a change in the set of pending entities that the block stores. The signal value is the number of pending entities that the block stores.

Compatibility Considerations

If your models use the **pe** or **#pe** signal to control simulation behavior, perform computations, or return results, your models might behave differently. The table summarizes the behavioral changes most likely to affect your models.

Affected Blocks	Change in Behavior
<ul style="list-style-type: none"> • Event-Based Entity Generator • Infinite Server • N-Server • Output Switch • Single Server • Time-Based Entity Generator 	<p>The pe signal does not have a sample time hit to reflect that an entity departs the first time it tries to depart. Such an entity is not a pending entity because it does not fail to depart.</p> <p>The same information is true for #pe in blocks that offer this port.</p> <p>Example</p> <p>In V3.1 (R2010a), the next model produces a plot containing no points because no entity fails to depart from the server.</p>  <p>In V3.0 (R2009b), the same model produces a plot that shows sample time hits in the pe signal when entities depart.</p> 
<ul style="list-style-type: none"> • Event-Based Entity Generator • Infinite Server • N-Server 	<p>When pe reflects the departure or other removal of a pending entity, the sample time hit occurs after the pending entity is no longer in the block. In earlier versions, when the pending entity is the only pending entity in the block, the sample time hit occurs when the departure or other removal is imminent. The sample time</p>

Affected Blocks	Change in Behavior
<ul style="list-style-type: none"> • Output Switch • Single Server • Time-Based Entity Generator 	<p>hit occurs at the same simulation time, but in a different sequence compared to other simultaneous events.</p> <p>Example</p> <p>The model, whose fragment appears in the next figure, behaves differently in different versions. Suppose a pending entity is in the server when the gate opens. In V3.1 (R2010a), the entity advances and acquires an attribute value of 1 from the pe signal. In earlier versions, the pe signal changes from 1 to 0 before the entity departs. As a result, when the entity advances, it acquires an attribute value of 0.</p> 
<ul style="list-style-type: none"> • Event-Based Entity Generator • Time-Based Entity Generator 	<p>When you configure the block to produce an error if an entity fails to depart, the error situation does not cause a sample time hit in pe. In this configuration, the block cannot store any pending entities, so there is no storage action to cause a sample time hit in pe.</p> <p>You see the effect of this change if, after the error occurs, you examine pe in the workspace or in a plot.</p>
<ul style="list-style-type: none"> • Infinite Server • N-Server 	<p>If a pending entity departs and one or more pending entities remain in the block, the pe signal has a single sample time hit of 1. In earlier versions, in this situation, the signal has a sample time hit of 0 followed by a sample time hit of 1.</p> <p>Example</p> <p>In the next model, at $T = 0.5$, one pending entity departs from the server and one pending entity remains. In V3.1 (R2010a), pe has a sample time hit to indicate the departure of the pending entity. The value is 1 because the block still contains another pending entity. In earlier versions, pe has a sample time hit of 0 followed by a sample time hit of 1.</p>



Renaming of Parameter to Enable Pending Entity Signal

Blocks that have an optional **pe** signal output port rename the parameter that you use to enable the port. The name in V3.0 (R2009b) is **Status of pending entity departure** or **Status of pending entity**. The new name in V3.1 (R2010a) is **Pending entity present in block**. The affected blocks are:

- Event-Based Entity Generator
- Infinite Server
- N-Server
- Output Switch
- Single Server
- Time-Based Entity Generator

Expanded Options for Opening Release Gate

You can configure the Release Gate block to open upon each sample time hit of an input signal. Set the **Open gate upon** parameter to **Sample time hit from port ts**.

Blocks in Attributes Library Must Get or Set at Least One Attribute

These blocks no longer support a configuration in which the table in the dialog box is empty:

- Get Attribute
- Set Attribute

Parameters and Parameter Values Being Removed

Affected Block	Affected Parameter	What Happens When You Use the Parameter?	Use This Instead	Compatibility Considerations
<ul style="list-style-type: none"> • FIFO Queue • LIFO Queue • Priority Queue 	Status of pending entity departure is being removed.	<p>In legacy models in which the parameter is set to On, the corresponding pe signal output port is inactive.</p> <p>In the library block, the parameter is unavailable.</p>	Number of entities in queue	To update legacy models, set Status of pending entity departure to Off. For more information, see the technique in “Determining Whether a Queue Is Nonempty”. The technique yields similar, but not identical, information.
<ul style="list-style-type: none"> • FIFO Queue • LIFO Queue • Priority Queue 	Capacity must have a positive value.	Warns if you set the value to 0.	A positive value. Alternatively, remove the block.	Remove queue blocks whose capacity is zero.

Version 3.0 (R2009b) SimEvents Software

This table summarizes what's new in Version 3.0 (R2009b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports Includes fixes	No

Support for Batch Simulation Using Rapid Simulation Target

SimEvents blocks support code generation using the Rapid Simulation target. You can now perform these tasks:

- “Accelerating Discrete-Event Simulations Using Rapid Simulation”
- “Varying Parameters Between Rapid Simulation Runs”
- “Sharing Executables for Discrete-Event Simulations”

The Batch Discrete-Event Simulations Using Rapid Simulation Target demo illustrates this feature by varying parameters between simulation runs.

This feature requires Real-Time Workshop® software and uses the Rapid Simulation target.

Expanded Options for Resetting Entity Departure Counter

The Entity Departure Counter block offers you more options for resetting the entity count during the simulation. The new options and corresponding values of the **Reset counter upon** parameter are listed in the table.

Option	Value of “Reset counter upon” Parameter
Reset counter upon each sample time hit of an input signal	Sample time hit from port ts
Reset counter upon each function call	Function call from port fcn

Version 2.4 (R2009a) SimEvents Software

This table summarizes what's new in Version 2.4 (R2009a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes Summary	Bug Reports Includes fixes	No

New features and changes introduced in this version are described here:

Debugger Supports Stepping, Breakpoints, and Querying

The new SimEvents debugger lets you use MATLAB functions to suspend a simulation at each step or breakpoint, and query simulation state to assess behavior. The debugger includes these functions:

<code>help</code>	Display help for debugger functions
<code>se_getdbopts</code>	SimEvents debugger options structure
<code>sedb.bdelete</code>	Delete breakpoints in discrete-event simulation
<code>sedb.blkinfo</code>	Block information in discrete-event simulation
<code>sedb.blklist</code>	Blocks and their identifiers in discrete-event simulation
<code>sedb.breakpoints</code>	List breakpoints in discrete-event simulation
<code>sedb.cont</code>	Continue simulation until next breakpoint
<code>sedb.currenttop</code>	Current operation in discrete-event simulation
<code>sedb.detail</code>	Customize debugger display in discrete-event simulation

<code>sedb.disable</code>	Disable breakpoints in discrete-event simulation
<code>sedb.enable</code>	Enable breakpoints in discrete-event simulation
<code>sedb.eninfo</code>	Entity information in discrete-event simulation
<code>sedb.evbreak</code>	Set breakpoint for execution or cancelation of event
<code>sedb.evcal</code>	Event calendar of discrete-event simulation
<code>sedb.evinfo</code>	Event information in discrete-event simulation
<code>sedb.gceb</code>	Name of currently executing block in discrete-event simulation
<code>sedb.gcebid</code>	Identifier of currently executing block in discrete-event simulation
<code>sedb.gcen</code>	Identifier of entity currently undergoing operation
<code>sedb.gcev</code>	Identifier of current event
<code>sedb.quit</code>	Quit discrete-event simulation debugging session
<code>sedb.runtoend</code>	Run until end of discrete-event simulation
<code>sedb.simtime</code>	Current time in discrete-event simulation
<code>sedb.step</code>	Single step in discrete-event simulation
<code>sedb.tbreak</code>	Set timed breakpoint in discrete-event simulation
<code>sedebg</code>	Debug discrete-event simulation

For more information, see these resources:

- “Overview of the SimEvents Debugger” in the SimEvents user guide documentation
- A video tutorial on the Web, in two parts:

- Basic Single Stepping and Querying
- Breakpoints and Advanced Querying
- “Building a Simple Discrete-Event Model” in the SimEvents getting started documentation (“Exploring the D/D/1 System Using the SimEvents Debugger” section)
- “Building a Simple Hybrid Model” in the SimEvents getting started documentation (“Confirming Event-Based Behavior Using the SimEvents Debugger” section)

Event Logging Options Removed from Configuration Parameters Dialog Box

The SimEvents pane of the Configuration Parameters dialog box no longer contains event logging options. The behavior of the event logging options in earlier versions is like the behavior of the new debugger.

Compatibility Considerations

The debugger behavior produces slightly different information, with a different format, compared to the information produced by the event logging parameters. The closest approximations to the previous behavior use the `detail` and `runtoend` functions of the debugger.

Event Logging Parameter Removed	Similar Behavior in Debugger
Display events in event calendar	<ol style="list-style-type: none"> 1 At the MATLAB command prompt, start the debugger on the model called <code>model</code>: <pre style="margin-left: 40px;">sedebug(model)</pre> 2 At the <code>sedebug>></code> prompt, configure the debugger and run the simulation until the end: <pre style="margin-left: 40px;">detail none detail('ev',1,'cal',1) runtoend</pre>

Event Logging Parameter Removed	Similar Behavior in Debugger
<p data-bbox="391 335 632 392">Log events when executed</p> <hr/> <p data-bbox="391 413 632 470">Log events when scheduled</p>	<p data-bbox="713 366 1297 423">1 At the MATLAB command prompt, start the debugger on the model called <code>model</code>:</p> <pre data-bbox="777 458 980 482">sedebug(model)</pre> <p data-bbox="713 527 1322 585">2 At the <code>sedebug>></code> prompt, configure the debugger and run the simulation until the end:</p> <pre data-bbox="777 619 980 708">detail none detail('ev',1) runtoend</pre> <p data-bbox="709 753 1311 874">The resulting log includes both execution messages and scheduling messages. Execution messages are not indented; scheduling messages are indented.</p>
<p data-bbox="391 894 617 986">Log entities advancing from block to block</p>	<p data-bbox="713 925 1297 982">1 At the MATLAB command prompt, start the debugger on the model called <code>model</code>:</p> <pre data-bbox="777 1017 980 1041">sedebug(model)</pre> <p data-bbox="713 1086 1322 1144">2 At the <code>sedebug>></code> prompt, configure the debugger and run the simulation until the end:</p> <pre data-bbox="777 1178 980 1267">detail none detail('en',1) runtoend</pre> <p data-bbox="709 1312 1288 1404">The resulting log includes entity advancement messages and other information. Entity advancement messages appear indented.</p>

To approximate the behavior that results from setting multiple parameters for the same model, you can concatenate input arguments in the `detail(...)`

command. For example, `detail('ev',1,'en',1)` is like logging event scheduling, event execution, and entity operations.

Discrete Event Signal to Workspace Block Clarifies Timing

You can no longer place the Discrete Event Signal to Workspace block in an atomic subsystem. The atomic subsystem executes at time instants that conflict with the time instants at which the event-based block in the subsystem executes. In earlier versions, placing the Discrete Event Signal to Workspace block in an atomic subsystem can produce unexpected results. Function-Call Subsystem and Enabled Subsystem are examples of atomic subsystems.

Compatibility Considerations

If your legacy model includes a Discrete Event Signal to Workspace block in an atomic subsystem, update the model as follows:

- 1** Move the Discrete Event Signal to Workspace block outside the atomic subsystem.
- 2** Connect the block to an output signal from the subsystem.

Version 2.3 (R2008b) SimEvents Software

This table summarizes what's new in Version 2.3 (R2008b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports Includes fixes	No

New features and changes introduced in this version are described here:

New Demos for Modeling Architectures and Manufacturing Processes

Version 2.3 (R2008b) introduces these new demonstrations:

Tutorial Demos

- Translating Events to Functions Calls


Advanced Technique Demos

- Asynchronous Execution of a Stateflow® Chart
- Building an Arrival Rate Estimator
- Interfacing with External File Formats
- Resource Allocation from Multiple Pools

Application Demos

- FIFO Buffer: Functional Model
- FIFO Buffer: Architectural Model
- Anti-Lock Braking System (ABS) Overview
- Batch Production Process
- Kanban Production System

Attribute Name Incrementing in Set Attribute and Get Attribute Blocks

The Add button  in the Set Attribute and Get Attribute blocks adds an attribute named `AttributeN` where N is a positive integer. In earlier versions, the button always adds an attribute named `Attribute1`.

Change in Parameter Name of Event-Based Entity Generator Block

The **Type of value change** parameter in the Event-Based Entity Generator block is now called **Type of change in signal value**. The new name is consistent with other blocks that have a parameter by that name.

Version 2.2 (R2008a) SimEvents Software

This table summarizes what's new in Version 2.2 (R2008a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports Includes fixes	No

New features and changes introduced in this version are

- “Initial Value Block in Signal Management Library” on page 20
- “Discrete Event Subsystem Supports Complex and Nonscalar Values” on page 20
- “Seed Management for Random Number Generators” on page 21
- “Configuration Parameters for Diagnostics” on page 21
- ““What’s This?” Context-Sensitive Help Available for Simulink Configuration Parameters Dialog” on page 21
- “New Demos” on page 22

Initial Value Block in Signal Management Library

The new Initial Value block is in a new library called Signal Management. This block sets a signal value before the first event occurs.

Also, the Signal Latch block has moved from the Gates library to the new Signal Management library.

Discrete Event Subsystem Supports Complex and Nonscalar Values

In Version 2.2 (R2008a), input signals to the Discrete Event Subsystem block can be real or complex signals of any dimension. In earlier versions, input signals to the block must be real scalars. For details about discrete event subsystems, see “Working with Discrete Event Subsystem Blocks”.

Seed Management for Random Number Generators

New functions and diagnostics help you ensure uniqueness of seeds of random number generators and manage sets of seeds in a series of simulation runs. For details, see these sections:

- “Varying Simulation Results by Managing Seeds”
- `se_getseeds` function reference page
- `se_setseeds` function reference page
- `se_randomize seeds` function reference page
- “Identical seeds for random number generators”

Configuration Parameters for Diagnostics

The Configuration Parameters dialog box has a new SimEvents Diagnostics pane to advise you of race conditions and help you manage seeds of random number generators.

For more information, see “SimEvents Diagnostics Pane”.

“What’s This?” Context-Sensitive Help Available for Simulink Configuration Parameters Dialog

R2008a introduces “What’s This?” context-sensitive help for parameters that appear in the Simulink Configuration Parameters dialog. This feature provides quick access to a detailed description of the parameters, saving you the time it would take to find the information in the Help browser.

To use the “What’s This?” help, do the following:

- 1 Place your cursor over the label of a parameter.
- 2 Right-click. A **What’s This?** context menu appears.

For example, the following figure shows the **What’s This?** context menu appearing after a right-click on the **Start time** parameter in the **Solver** pane.



- 3 Click **What's This?** A context-sensitive help window appears showing a description of the parameter.

New Demos

SimEvents software Version 2.2 (R2008a) introduces these new demos:

- Managing Race Conditions
- Avoiding Identical Seeds for Random Number Generators
- Seed Management Workflow for Random Number Generators
- Asynchronous Clock Domains
- Rate-Based Shared Processor

Also, the demo formerly named “Explicit Routing for Distributed Processing” is now called “Distributed Processing for Multi-Class Jobs.”

Version 2.1 (R2007b) SimEvents Software

This table summarizes what's new in Version 2.1 (R2007b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes	No

New features and changes introduced in this version are

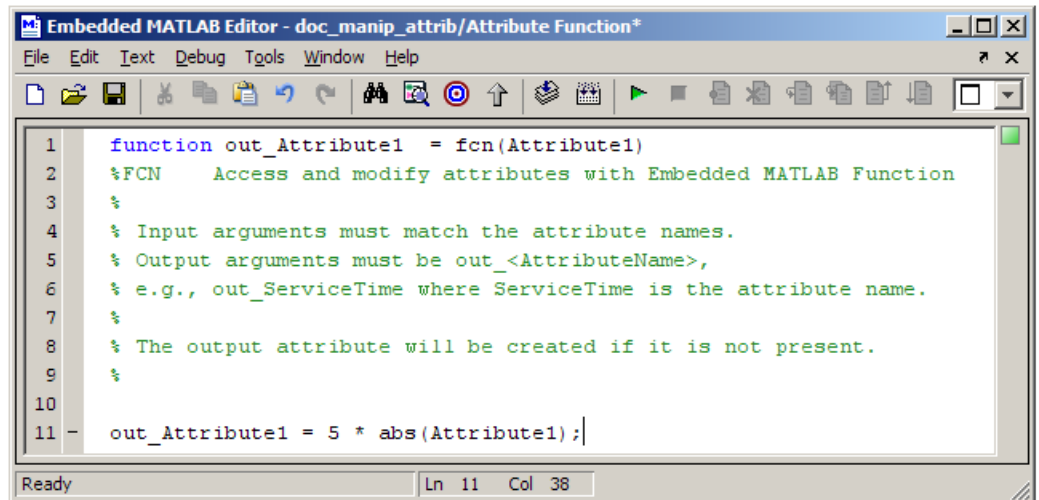
- “Attribute Computations Using Embedded MATLAB Code” on page 23
- “Attributes Support Complex Values” on page 25
- “Enhanced Visibility and Logging of Events” on page 25
- “New Demos for Shared-Resource Applications and Advanced Techniques” on page 26
- “Subsystem Connection Port for Entity Paths” on page 28
- “Configuration Parameters to Control Livelock” on page 28
- “Processing Events Via the Event Calendar Instead of Immediately” on page 29
- “Enhanced Support for Multiple Simultaneous Transitions in Switches and Gate” on page 31
- “Change in Indexing in Attribute Scope Block” on page 33

Attribute Computations Using Embedded MATLAB Code

The Attribute Function block lets you conveniently set and modify attributes using Embedded MATLAB® code. For details, see “Writing Functions to Manipulate Attributes”.

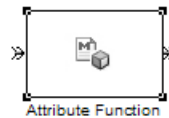
Simplifying a Model Using the Attribute Function Block

The following figures indicate recommended ways to multiply the absolute value of an attribute by a constant in SimEvents software Version 2.1 (R2007b) and earlier versions. The earlier version is more complicated because of necessary steps to ensure correct timing. By contrast, the Attribute Function block ensures correct timing automatically.

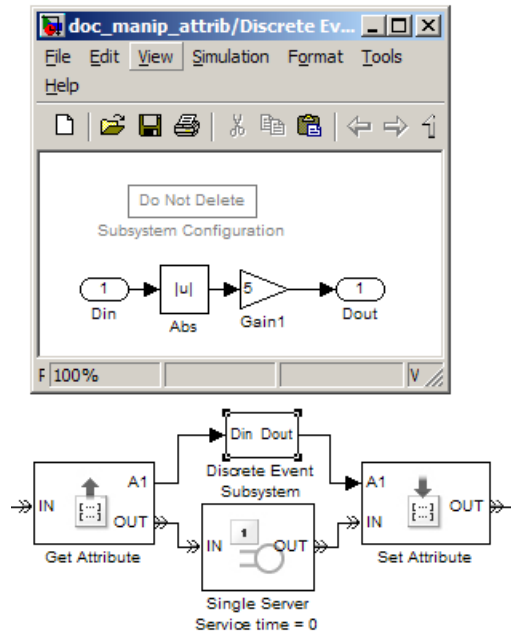


```
Embedded MATLAB Editor - doc_manip_attrib/Attribute Function*
File Edit Text Debug Tools Window Help
[Icons]
1 function out_Attribute1 = fcn(Attribute1)
2 %FCN Access and modify attributes with Embedded MATLAB Function
3 %
4 % Input arguments must match the attribute names.
5 % Output arguments must be out_<AttributeName>,
6 % e.g., out_ServiceTime where ServiceTime is the attribute name.
7 %
8 % The output attribute will be created if it is not present.
9 %
10
11 out_Attribute1 = 5 * abs(Attribute1);
```

Ready Ln 11 Col 38



Manipulating Attribute Value Using Attribute Function Block



Manipulating Attribute Value in Earlier Versions

Attributes Support Complex Values

In Version 2.1 (R2007b), attributes can assume complex values, not only real values.

Enhanced Visibility and Logging of Events

In Version 2.1 (R2007b), SimEvents software changes the set of events that appear in event logs:

- Event logs show a new kind of event, called an *entity request* event. This event is a notification that an entity input port of a block has become available. To understand the name entity request event, think of the block as requesting an entity from a preceding block. For example, upon becoming empty, a single server requests an entity from a preceding block. A preceding block's response to the notification might result in an entity advancement.

In earlier versions, entity request events do not appear in event logs.

- Event logs show a new kind of event, called a *storage completion* event. This event exists only in an Output Switch block with the **Store entity before switching** parameter selected. When an entity arrives at the block, the block schedules a storage completion event at the current simulation time. Upon execution of the storage completion event, the block determines whether the entity can advance to a subsequent block.

In earlier versions, storage completion events do not appear in event logs.

- Event logs always show the events listed in the following table, regardless of how you set the **Resolve simultaneous signal updates according to event priority** parameter in the corresponding blocks. This parameter determines whether the event priority is a number you specify in the block dialog box or a system-level category denoted by SYS1.

Events That Affect Entities and Are Caused By Signal-Based Events

Event	Block
Entity generation	Event-Based Entity Generator
Counter reset	Entity Departure Counter
Gate event (gate opening or gate closing)	Enabled Gate
Release	Release Gate
Port selection	Input Switch, Output Switch, Path Combiner

In earlier versions, event logs show these events only if you select **Resolve simultaneous signal updates according to event priority** in the block dialog box.

Also, event logs and entity logs in Version 2.1 (R2007b) are more readable and contain hyperlinks that highlight the corresponding blocks.

New Demos for Shared-Resource Applications and Advanced Techniques

SimEvents software Version 2.1 (R2007b) introduces these new demonstration models:

Tutorial Demos

- Server Blocks and Service Time
- Input and Output Switching
- Schedule Timeout and Cancel Timeout Blocks

Advanced Technique Demos

- Buffering for Variable-Size Messages
- Delayed Function Calls in Pulse Width Modulation
- Explicit Routing for Distributed Processing
- Markov-Modulated Poisson Process
- Queue with Flushing Capability
- Variable Entity Replication
- Variable-Capacity Queue

Application Demos

- Shared Communication Buffer Management
- Processor Sharing Via Time Slicing

Consolidation and Removal of Some Tutorial Demos

The new Server Blocks and Service Time demo replaces these earlier demos:

- Service Time from Attribute
- Specifying Service Time in Single Server
- Specifying Service Time in Infinite Server Block
- Single Server Block Versus Infinite Server Block

The new Input and Output Switching demo replaces these earlier demos:

- Input Switching Using Signal
- Output Switching Using Signal

Changes in Categorization, Titles, and Content of Some Demos

SimEvents demos have been recategorized in the Help browser. Some demos have changed their titles or content.

Title in Version 2.1 (R2007b)	Title in Earlier Versions
Task Sharing with Two Levels of Priority and Preemption	Preemptive Operating System with Two Levels of Priority
Multitasking with Dependent Tasks	Multitasking Model with Dependent Tasks
Operating System with Prioritized Task Execution	Operating System Model with Prioritized Task Execution
Entity Combiner for Assembling Components (with simpler design using the Entity Combiner and Attribute Function blocks)	Aggregation: Assembling a Vehicle Chassis

Also, the G/G/1 Queuing System and Little's Law demo has a simpler design using the Attribute Function and Embedded MATLAB Function blocks.

Subsystem Connection Port for Entity Paths

The Conn block represents an entity input port or entity output port in a virtual subsystem. The model window's **Edit > Create Subsystem** menu option automatically creates connection ports. Copying the Conn block from its library is a convenient way to add more entity ports to an existing subsystem.

Configuration Parameters to Control Livelock

The SimEvents pane of the Configuration Parameters dialog box offers new parameters for setting thresholds related to livelock. Also, the **Execution order of simultaneous events** parameter has been renamed **Execution order**.

New Parameter	Description
Maximum events per block	Limit the number of entity generation, service completion, subsystem execution, and function-call events that each SimEvents block performs at each fixed time instant.
Maximum events per model	Limit the total number of events scheduled via the event calendar at each fixed time instant.

For more information, see “Livelock Prevention” or the configuration parameter descriptions.

Processing Events Via the Event Calendar Instead of Immediately

In Version 2.1 (R2007b), SimEvents software changes its processing of each event in the next table when you do not select **Resolve simultaneous signal updates according to event priority** in the corresponding block. In this case, the event has a system-level event priority denoted by SYS1, and the application processes the event via the event calendar. Using the event calendar decouples the scheduling and the execution of events. “Event Sequencing” describes how the application processes multiple simultaneous events.

Events That Affect Entities and Are Caused By Signal-Based Events

Event	Block
Entity generation	Event-Based Entity Generator
Counter reset	Entity Departure Counter
Gate event (gate opening or gate closing)	Enabled Gate
Release	Release Gate
Port selection	Input Switch, Output Switch, Path Combiner

Also, each entity request event has a system-level priority denoted by SYS2, and the application processes the event via the event calendar.

In earlier versions, the application applies “immediate” processing for entity requests by storage blocks, as well as for events in the table when you do not select the **Resolve simultaneous signal updates according to event priority** parameter in the corresponding block.

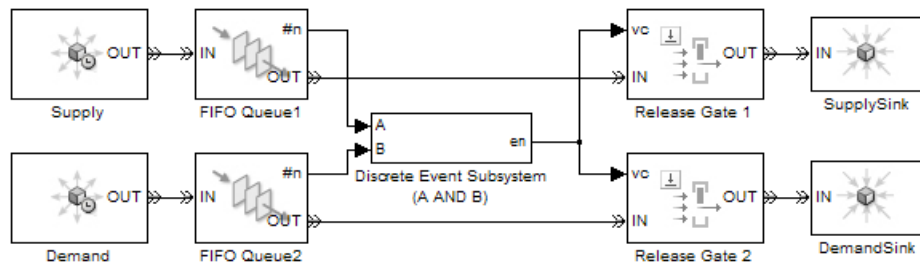
For details about supported events and the processing of simultaneous events, see “Working with Events” and “Managing Simultaneous Events”.

Compatibility Considerations

Most models are unaffected by the change in event processing. However, some models might behave differently, because events processed immediately in earlier versions are deferred to the event calendar in SimEvents software Version 2.1 (R2007b). Models that behave differently tend to involve cycles in simulation processing and cascades of simultaneous events (for example, an event has multiple consequences that occur at time T , each of which has further consequences also at time T).

Example Showing Change in Behavior

The model below attempts to simultaneously advance one entity from each queue, whenever both queues are nonempty.



Suppose that the top queue contains one entity and an entity arrives at the previously empty bottom queue. Assuming that no block in the model has its **Resolve simultaneous signal updates according to event priority** parameter selected, the entity arrival has the following cascade of consequences:

- 1 The bottom queue updates its **#n** output signal to 1.

- 2 The discrete event subsystem evaluates the condition (A AND B) and returns a value of 1. The previous value of this signal was 0.
- 3 Each of the two Release Gate blocks detects a value-change event at its **vc** signal input port. In SimEvents software Version 2.1 (R2007b), each of the two gates schedules a release event on the event calendar.
- 4 One gate opens, which has these consequences:
 - a An entity advances from the corresponding queue to the sink.
 - b The corresponding queue updates its **#n** output signal to 0.
 - c The discrete event subsystem reevaluates the condition (A AND B) and returns a value of 0.
- 5 In Version 2.1 (R2007b), the other gate opens, which has these consequences:
 - a An entity advances from the corresponding queue to the sink.
 - b The corresponding queue updates its **#n** output signal to 0.
 - c The discrete event subsystem reevaluates the condition (A AND B) and returns a value of 0.

In earlier versions, the gates do not schedule release events on the event calendar if the corresponding **Resolve simultaneous signal updates according to event priority** parameter is not selected. As a result, step 4c negates the value-change event at the other gate and step 5 does not occur. This example involves cycles in simulation processing, because an event at the gate affects the value of the **#n** signal of a preceding block. This example involves cascades of simultaneous events, because the new value of 1 for the condition (A AND B) causes two release events, each of which causes the condition (A AND B) to assume the value 0.

Enhanced Support for Multiple Simultaneous Transitions in Switches and Gate

The blocks in this table model the effects of all transitions in their input signals, even if multiple transitions occur simultaneously.

Block	Input Signal
Enabled Gate	en
Input Switch	p
Output Switch	p
Path Combiner	p

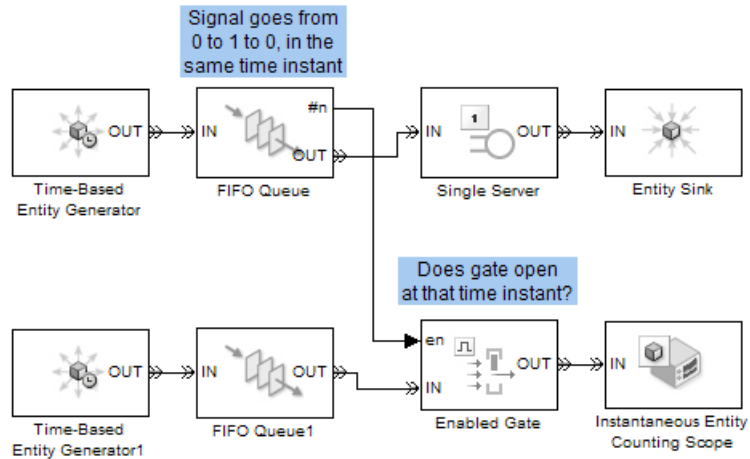
In earlier versions of SimEvents software, selecting the **Resolve simultaneous signal updates according to event priority** option causes the blocks to model only the last transition at a given value of the simulation clock.

Compatibility Considerations

The behavior of some simulations changes depending on whether the application models intermediate transitions in an **en** or **p** input signal in the blocks in the table above.

Example Showing Change in Behavior

In the model below, the **en** signal transitions from 0 to 1 and then from 1 to 0 in the same time instant. Earlier versions model only the latter transition, so the gate does not open. Version 2.1 (R2007b) models both transitions, so the gate opens and then closes in the same time instant.



Change in Indexing in Attribute Scope Block

When you set the **X** value from parameter to Index in the Attribute Scope block, it creates a plot that reflects 1-based indexing. That is, the first entity corresponds to a data point whose value on the horizontal axis is 1. In earlier versions, the plot reflects 0-based indexing.

Version 2.0 (R2007a) SimEvents Software

This table summarizes what's new in Version 2.0 (R2007a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes	No

New features and changes introduced in this version are

- “Attributes Support Multidimensional Values” on page 34
- “Combining and Splitting Entities” on page 35
- “Timeout Feature Establishes Entity Time Limits” on page 35
- “New Demos for Video Processing, Communications, and Architecture Modeling” on page 37
- “Output Switch Block Options for Storage and Initial Condition” on page 37
- “Entity Departure Counter Block Can Create Attribute” on page 39
- “Changes in Names of Parameters Related to Event Priorities” on page 40
- “Change in Default Entity Type of Entity Generators” on page 42
- “Obsolete Blocks” on page 42

Attributes Support Multidimensional Values

Version 2.0 (R2007a) introduces new versions of the Get Attribute and Set Attribute blocks in a new Attributes library. The new blocks offer these enhancements compared to the earlier versions:

- Attributes can assume values that are vectors, matrices, or multidimensional arrays with up to 32 dimensions, not just scalars. This enhancement facilitates modeling dense payloads via attributes.

- Each instance of a Set Attribute block can assign an arbitrary number of attributes, and each instance of a Get Attribute block can retrieve an arbitrary number of attributes.
- The dialog boxes use a grid on a single tab, making it easier for you to see the entire list of attributes that a block sets or gets.

Compatibility Considerations

If your legacy models contain Get Attribute or Set Attribute blocks from the earlier library, those blocks continue to work in Version 2.0 (R2007a). However, the blocks are considered obsolete, as described in “Obsolete Blocks” on page 42.

Combining and Splitting Entities

The new Entity Combiner block lets you combine entities, analogous to combining components to create a larger whole. The block provides options for managing information (attributes and timers) associated with the component entities, so you can think of the operation as bundling the information that entities carry with them.

You can configure the Entity Combiner block to make the combining operation reversible via the Entity Splitter block.

The Entity Combiner and Entity Splitter blocks reside in the new Entity Management library.

Timeout Feature Establishes Entity Time Limits

You can model point-to-point timing constraints by limiting the amount of time an entity spends during the simulation on designated entity paths. Exceeding the limit causes the entity to depart immediately from the storage block where it resides, such as a queue, when the clock reaches the time limit. To learn how to use this feature, see “Forcing Departures Using Timeouts” in the SimEvents user guide documentation.

The timeout feature involves new blocks, as well as new parameters in existing blocks.

New Block	Purpose
Schedule Timeout	Schedule timeout event for each entity
Cancel Timeout	Cancel timeout event for each entity

Existing Blocks with New Timeout-Related Parameters

- FIFO Queue
- LIFO Queue
- Priority Queue
- Infinite Server
- N-Server
- Single Server
- Output Switch

New Parameter of Existing Blocks	Purpose
Enable TO port for timed-out entities on Timeout tab	Provide a TO entity output port through which an entity departs upon timing out
Number of entities timed out on Statistics tab	Output a signal, #to , that indicates the number of entities that have timed out from the block since the start of the simulation

Compatibility Considerations

If you save a model containing a queue, server, or Output Switch block using V2.0 (R2007a), then opening the model in V1.2 (R2006b) produces warnings like these:

```
Warning: In instantiating linked block 'mysys/FIFO Queue' :
FIFO Queue block (mask) does not have a parameter named
'EnableTOPort'.
Warning: In instantiating linked block 'mysys/FIFO Queue' :
```


FIFO Queue block (mask) does not have a parameter named 'StatNumberTimedout'.

Saving the model in the earlier version prevents the warnings from reappearing, but causes the block to omit timeout-related ports and behavior if you later open the model in V2.0 (R2007a).

New Demos for Video Processing, Communications, and Architecture Modeling

Version 2.0 (R2007a) introduces these new demonstration models:

Tutorial Demos

- Transporting Multidimensional Data Using Attributes
- Packet Creation, Transmission and Error Analysis

Application Demos

- Distributed Video Processing
- Distributed Processing Resource Modeling
- Video Streaming Over Bandwidth-Limited Communication Channel
- Bit Timing Recovery Using Fixed-Rate Resampling and SimEvents

Change in ARQ Demo

The Selective-Repeat Automatic Repeat Request demo reverses the interpretation of the CRC check compared to V1.2 (R2006b). The interpretation now matches that of the similar Go-Back-N Automatic Repeat Request demo. In V2.0 (R2007a), both demos use a CRC check value of 1 to correspond to an ACK message.

Output Switch Block Options for Storage and Initial Condition

The Output Switch block offers enhancements that can prevent the need for additional blocks to set initial conditions or to prevent latency. The new parameters apply to signal-based switching and are available only when you set **Switching criterion** to From signal port p. The new parameters are

in the table below. For details, see “Output Switching Based on a Signal” and the block’s reference page.

New Parameter	Purpose
Specify initial port selection	Determine whether the block uses an initial port selection from the dialog box.
Initial port selection	The entity output port that the block selects when the simulation begins. The block uses this value instead of the p signal until the signal has its first sample time hit.
Store entity before switching	If you select this option, the block can store one entity at a time. Furthermore, the block decouples its arrival and departure processing to give other blocks in the simulation an opportunity to update the p signal if appropriate. If you do not select this option, the block processes an arrival and departure as an atomic operation and assumes that the p signal is already up to date at the given time.
Status of pending entity on Statistics tab	Output a signal, pe , that indicates when the block stores an entity after trying and failing to output it. A value of 0 indicates when the storage location is empty.

For other changes in this release that affect parameters of the Output Switch block, see “Timeout Feature Establishes Entity Time Limits” on page 35 and “Changes in Names of Parameters Related to Event Priorities” on page 40.

Compatibility Considerations

In some cases, the block enhancements let you optionally simplify models that you do not need to share with users of earlier versions:

- If your model precedes an Output Switch block with a Signal Latch block to create an initial condition for the **p** signal, and if the **p** signal does not branch to become an input for another block, then you can remove

the Signal Latch block and instead use the new **Specify initial port selection** option in the switch block.

- If your model precedes an Output Switch block with a Single Server block whose **Service time** parameter is zero and whose sole purpose was to ensure an up-to-date **p** signal, then you can remove the Single Server block and instead use the new **Store entity before switching** option in the switch block.

If you save a model containing an Output Switch block using V2.0 (R2007a), then opening the model in V1.2 (R2006b) produces warnings like these:

```
Warning: In instantiating linked block 'mysys/Output Switch' :
Output Switch block (mask) does not have a parameter named
'InitialConditionsOpt'.
```

```
Warning: In instantiating linked block 'mysys/Output Switch' :
Output Switch block (mask) does not have a parameter named
'InitialConditions'.
```

```
Warning: In instantiating linked block 'mysys/Output Switch' :
Output Switch block (mask) does not have a parameter named
'EntityBufferOpt'.
```

```
Warning: In instantiating linked block 'mysys/Output Switch' :
Output Switch block (mask) does not have a parameter named
'StatPendingEntity'.
```

Saving the model in the earlier version prevents the warnings from reappearing, but causes the block to omit ports and behavior related to the V2.0 (R2007a) enhancements if you later open the model in V2.0 (R2007a).

Entity Departure Counter Block Can Create Attribute

If you configure the Entity Departure Counter block to write the count to an attribute, then you can select the new **Create attribute if not present** parameter to have the block create the attribute if it does not already exist. The block then sets the value of the attribute according to the entity count.

In earlier versions, the block sets the value of the attribute but does not create it.

Changes in Names of Parameters Related to Event Priorities

Parameters related to optional priorities of events have been renamed to be more suggestive of how the option works. The name **Resolve simultaneous signal updates according to event priority** replaces names that start with **Specify event priority**. In a subset of affected blocks, the name **Event priority** replaces similar names. For more information about what the parameters mean, see “Choosing How to Resolve Simultaneous Signal Updates”.

The table below itemizes the blocks and parameters that have changed.

Block	Parameter Name in V1.2 (R2006b)	Parameter Name in V2.0 (R2007a)
Discrete Event Inport	Specify event priority for executing subsystem	Resolve simultaneous signal updates according to event priority
	Subsystem execution event priority	Event priority
Enabled Gate	Specify event priority for gate opening and closing	Resolve simultaneous signal updates according to event priority
Entity Departure Counter	Specify event priority for counter reset	Resolve simultaneous signal updates according to event priority
Event-Based Entity Generator	Specify event priority for entity generation	Resolve simultaneous signal updates according to event priority
	Generation event priority	Event priority

Block	Parameter Name in V1.2 (R2006b)	Parameter Name in V2.0 (R2007a)
Input Switch	Specify event priority for port selection	Resolve simultaneous signal updates according to event priority
Output Switch	Specify event priority for port selection	Resolve simultaneous signal updates according to event priority
Path Combiner	Specify event priority for port precedence selection	Resolve simultaneous signal updates according to event priority
Release Gate	Specify event priority for gate opening	Resolve simultaneous signal updates according to event priority
Signal Latch	Specify event priority for writing to memory	Resolve simultaneous signal updates according to event priority on Write tab
	Specify event priority for reading from memory	Resolve simultaneous signal updates according to event priority on Read tab
Signal-Based Event to Function-Call Event	Specify event priority for function-call generation	Resolve simultaneous signal updates according to event priority
	Function-call event priority	Event priority

Block	Parameter Name in V1.2 (R2006b)	Parameter Name in V2.0 (R2007a)
Signal-Based Function-Call Event Generator	Specify event priority for function-call generation	Resolve simultaneous signal updates according to event priority
	Function-call event priority	Event priority

This change merely renames parameters and does not change the behavior of affected blocks.

Change in Default Entity Type of Entity Generators

The default value of **Entity type** in the Time-Based Entity Generator and Event-Based Entity Generator block is **Blank**. In earlier versions, the default value is **Standard**. This change in default value does not affect blocks in a saved model but only affects new instances of the block that you copy from the library to a model.

Obsolete Blocks

The table below indicates blocks that are obsolete as of the current version or that are planned to be removed in a future version.

Obsolete Block	Removed from Version	Replacement
Get Attribute block from simeventsattributes1 library	Future version	Get Attribute block from simeventsattributes2 library
Set Attribute block from simeventsattributes1 library	Future version	Set Attribute block from simeventsattributes2 library

Version 1.2 (R2006b) SimEvents Software

This table summarizes what's new in Version 1.2 (R2006b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes	No

New features and changes introduced in this version are

- “Event-Based Sequence Generator Block” on page 43
- “New Tutorial and Application Demos” on page 44
- “Event Translation Block Supports Delay from a Signal” on page 44
- “Routing Blocks Support Unlimited Entity Ports” on page 45
- “Initial Outputs of SimEvents Blocks” on page 46
- “History Options and Other Changes in Scope Blocks” on page 48
- “Parameters for Lognormal Distribution” on page 49
- “SimEvents Blocks Compatible with Accelerator Mode” on page 50
- “Livelock Detection” on page 50

Event-Based Sequence Generator Block

The new Event-Based Sequence block provides data to an event-driven process by producing a scalar event-based output signal whose values come from a vector. The block selects the next value from the vector upon each notification from a port of a subsequent block. For example, if you connect the Event-Based Sequence block to the **t** input port of a Single Server block, the values in the vector become the service times for the entities arriving at the server. You provide the values in the vector, but do not need to know in advance when the entities arrive at the server.

New Tutorial and Application Demos

Version 1.2 (R2006b) introduces these new demonstration models:

Tutorial Demos

- Entity Combiner for Assembling Components
- Task Sharing with Two Levels of Priority and Preemption
- Multitasking with Dependent Tasks

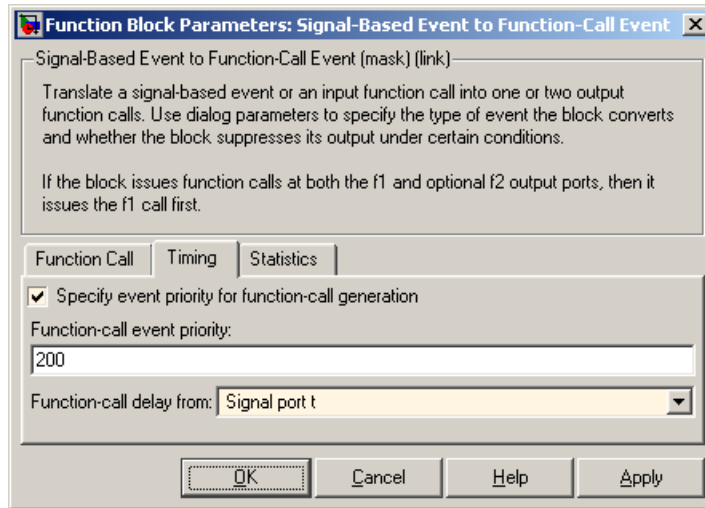
Application Demos

- Go-Back-N Automatic Repeat Request
- Operating System with Prioritized Task Execution
- Ethernet Local Area Network
- Anti-Lock Braking System (ABS) Model
- Anti-Lock Braking System (ABS) Model with Queuing Delay
- Anti-Lock Braking System (ABS) Model Using CAN Communications

Event Translation Block Supports Delay from a Signal

The Signal-Based Event to Function-Call Event block can delay its generation of a function call by an amount of time that you specify using either an input signal or the **Function-call time delay** parameter. In V1.1 (R2006a), the block lets you specify the delay amount using the parameter, but not an input signal.

To access the new feature, select **Specify event priority for function-call generation** (or, in V2.0 (R2007a), select **Resolve simultaneous signal updates according to event priority**). Then set the new **Function-call delay from** parameter to **Signal port t**, as shown. Then connect a nonnegative-valued signal to the **t** signal input port that appears on the block.



Compatibility Considerations

If you save a model containing the Signal-Based Event to Function-Call Event or Discrete Event Subsystem block using V1.2 (R2006b), then opening the model in V1.1 (R2006a) produces warnings like these:

```
Warning: In instantiating linked block 'mysys/Signal-Based
Event to Function-Call Event' : Signal-Based Event to Function-
Call Event block (mask) does not have a parameter named
'FunctionCallDelayFrom'.
```

Saving the model in the earlier version prevents the warnings from reappearing, but causes the Signal-Based Event to Function-Call Event block to omit the **t** input port if you later open the model in V1.2 (R2006b).

Routing Blocks Support Unlimited Entity Ports

The **Number of entity input ports** parameter of the Input Switch and Path Combiner blocks can be any positive integer. The **Number of entity output ports** parameter of the Output Switch and Replicate blocks also can be any positive integer. In V1.1 (R2006a), these parameters can assume only the values 1, 2, 3, and 4.

Compatibility Considerations

If you save a model in which one of the blocks listed above has more than four entity input ports or more than four entity output ports, then the model will not work in V1.1 (R2006a).

Initial Outputs of SimEvents Blocks

All SimEvents blocks now have well-defined initial values for any numerical output signals they produce.

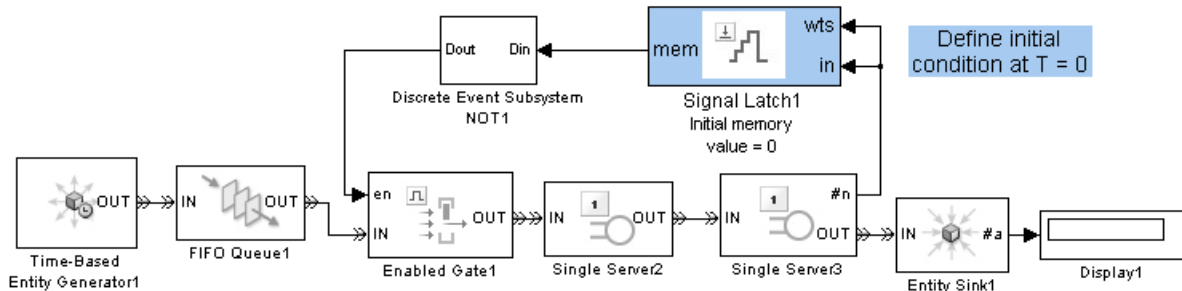
The initial value of an output signal of a SimEvents block is in effect from the start of the simulation until the block updates the output signal for the first time during the simulation. For example, if an N-Server block is configured to produce a **#n** output signal representing the number of entities in the server, then **#n** has a well-defined initial value of 0 at the start of the simulation. The initial value persists until the first arrival of an entity at the N-Server block, which could occur well after the start of the simulation, if at all.

The block reference pages indicate the initial values of the block output signals.

Compatibility Considerations

If you connect the Signal Latch block to a **ts**, **tr**, or **vc** signal input port of a SimEvents block, the input port might detect an event at the start of the simulation in V1.1 (R2006a) that no longer occurs in V1.2 (R2006b). This is because the Signal Latch block assumes its initial condition in a true initialization stage in V1.2 (R2006b) rather than slightly after the simulation start in V1.1 (R2006a). If your model relies on an event at the start of the simulation (to invoke a discrete event subsystem or generate an event or an entity, for example), then you might see a change in simulation behavior between the two versions.

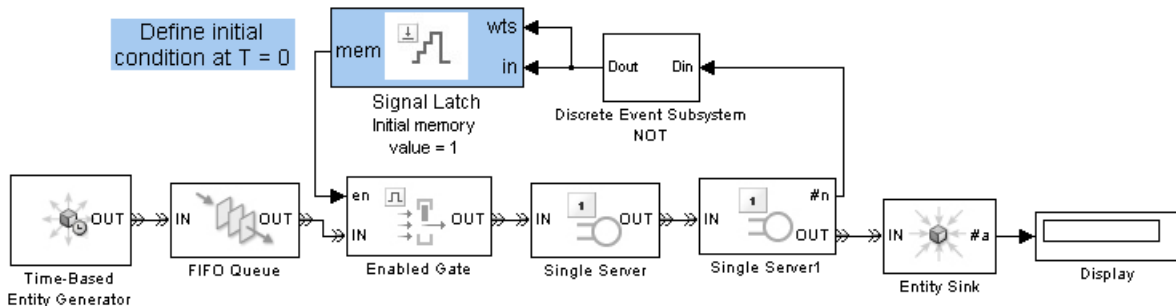
For example, the model below uses a Discrete Event Subsystem block to compute a signal that indicates whether a gate is open or closed.



Subsystem Invoked at Simulation Start in V1.1 (R2006a), but Not V1.2 (R2006b)

In V1.1 (R2006a), the Signal Latch block's output signal has a sample time hit at the start of the simulation. This sample time hit invokes the subsystem, which initializes the gate's **en** input signal to 1. As a result, the gate is open at the start of the simulation. In V1.2 (R2006b), the Signal Latch block does not have a sample time hit at the start of the simulation, so the initial condition of the subsystem's output determines the initial condition of the gate's **en** input signal. As a result, the gate is closed at the start of the simulation.

An alternative approach that works in both versions is to move the Signal Latch block so that it follows the Discrete Event Subsystem block. The Signal Latch block directly provides the gate's initial condition.



Correct Gate Initialization in Both V1.1 (R2006a) and V1.2 (R2006b)

History Options and Other Changes in Scope Blocks

The following blocks include new **Store data when scope is closed** and **Limit data points to** parameters on the new **Data History** tab of the dialog box:

- Attribute Scope
- Instantaneous Entity Counting Scope
- Instantaneous Event Counting Scope
- Signal Scope
- X-Y Attribute Scope
- X-Y Signal Scope

The parameters determine how much data the blocks cache, letting you balance data visibility with simulation efficiency. Caching data lets you view it later, even if the scope is closed during part or all of the simulation. Caching less or no data accelerates the simulation and uses less memory. In V1.1 (R2006a), if you have the scope closed for the first T seconds of simulation and then open the scope, you can view only the data for $t > T$.

Other Changes in Scope Blocks

Version 1.2 (R2006b) changes some aspects of the way you interact with the scope blocks:

- A Pan toolbar button lets you move your view of a plot.
- A Parameters toolbar button opens the block dialog box.
- Double-clicking on the block opens the plot if it is not already open. In V1.1 (R2006a), double-clicking on the block opens the block dialog box. To open the block dialog box in V1.2 (R2006b), click the Parameters toolbar button on the plot.
- The autoscale feature no longer changes the initial axis limits that you specify in the block dialog box. A new **Save axes limits** menu option lets you update the initial axis limits to match the plot's current limits. The current limits might differ from their initial values due to stretching, shifting, panning, zooming, or autoscaling operations that occurred since the initial values were last set.

- The former **Open at start of simulation** parameter is now called **Open scope at start of simulation** and has moved from the **Figure** tab of the dialog box to the **Plotting** tab.

The scope blocks also plot initial conditions without a plotting marker. In V1.1 (R2006a), initial conditions typically do not appear in plots.

Finally, the scope blocks run significantly faster in V1.2 (R2006b).

Compatibility Considerations

If your legacy models contain scope blocks that plot more than 1000 points, then the default values of the new **Store data when scope is closed** and **Limit data points to** parameters cause the scope to retain only the last 1000 points. To plot all points, set **Store data when scope is closed** to **Unlimited**.

If you save a model containing a scope block using V1.2 (R2006b), then opening the model in an earlier version produces warnings about the parameters that are not in the earlier block. For example,

```
Warning: In instantiating linked block 'mysys/Attribute  
Scope' : Attribute Scope block (mask) does not have a parameter  
named 'DataStoreOption'.
```

```
Warning: In instantiating linked block 'mysys/Attribute  
Scope' : Attribute Scope block (mask) does not have a parameter  
named 'DataPointsLimit'.
```

Saving the model in the earlier version prevents the warnings from reappearing, but also causes the block to use default values for the new parameters if you later open the model in V1.2 (R2006b).

Parameters for Lognormal Distribution

The Event-Based Random Number block produces random numbers from a lognormal distribution when you set the **Distribution** parameter to **Lognormal1**. Different texts use different parameterizations of the lognormal distribution. V1.2 (R2006b) renames some parameters in this block to clarify the relationship between a lognormal random variable X and the normal random variable $\log(X)$.

V1.1 (R2006a) Parameter Name	V1.2 (R2006b) Parameter Name
Scale	Mu
Shape	Sigma

Compatibility Considerations

The block behaves the same in V1.1 (R2006a) and V1.2 (R2006b) because the change merely renames parameters. However, the parameter names in V1.2 (R2006b) more accurately reflect the block's behavior.

SimEvents Blocks Compatible with Accelerator Mode

All SimEvents blocks are compatible with accelerator mode. Version 1.1 (R2006a) does not support simulating models in accelerator mode if the models contain the Event-Based Random Number block.

Livelock Detection

SimEvents software can detect livelock during a simulation. When it detects livelock, it halts the simulation with an error message that indicates too many simultaneous events. In V1.1 (R2006a), livelock can potentially cause MATLAB software to crash.

For details, see “Livelock Prevention”.

Compatibility Considerations

It is possible for the application to consider a situation to be livelock when it is actually a large but finite loop. Such simulations might work in V1.1 (R2006a) but not in V1.2 (R2006b).

Version 1.1 (R2006a) SimEvents Software

This table summarizes what's new in Version 1.1 (R2006a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports Includes fixes	No

New features and changes introduced in this version are described here.

Replicate Block Supports Partial Replication

The Replicate block supports partial replication and offers more flexibility when you choose complete replication. New parameters in the block's dialog box are in the table below.

Parameter	Description
Replicate entity when	Lets you choose whether the block accepts arriving entities for replication only when all entity output ports are not blocked or whenever at least one entity output port is not blocked. The first option is the default.
If an output port becomes blocked during replication	Determines how the block responds if a departure through one entity output port causes another entity output port to become blocked.
Number of entities departed	Toggles the optional output signal #d , representing the number of departed entities.

Compatibility Considerations

By default in V1.1 (R2006a), when a departure through one entity output port causes another entity output port to become blocked, the result is a discarded entity with no error or warning message. If this phenomenon occurs in your legacy models, then the result in V1.0 (R14SP3+) might be an error message or incorrect behavior. If you want to learn when this phenomenon occurs in your legacy models that you simulate using V1.1 (R2006a), then set **If an output port becomes blocked during replication** to either Warn and discard entity, or Error.

The default values of the other new parameters added in V1.1 (R2006a) are consistent with the block's behavior in V1.0 (R14SP3+), so legacy models need no changes to accommodate these new features.

If you save a model containing the Replicate block using V1.1 (R2006a), then opening the model in V1.0 (R14SP3+) produces warnings about the parameters that are not in the V1.0 block. For example,

```
Warning: In instantiating linked block 'mysys/Replicate' :  
  Replicate block (mask) does not have a parameter named  
  'ReplicateEntityWhen'.  
Warning: In instantiating linked block 'mysys/Replicate' :  
  Replicate block (mask) does not have a parameter named  
  'ActionUponBlocking'.  
Warning: In instantiating linked block 'mysys/Replicate' :  
  Replicate block (mask) does not have a parameter named  
  'StatNumberDeparted'.
```

Also, simulating that model under V1.0 causes the block to exhibit its V1.0 behavior, which is to omit a **#d** output signal and to replicate the arriving entity only when all entity output ports are not blocked. Saving the model in V1.0 prevents the warnings from reappearing in V1.0 but also causes the block to exhibit its V1.0 behavior if you later open the model in V1.1.

Version 1.0 (R14SP3+) SimEvents Software

This table summarizes what's new in Version 1.0 (R14SP3+):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Not applicable	Bug Reports	No

New features introduced in this version are described here.

Introduction to SimEvents

SimEvents software extends Simulink software with tools for modeling and simulating discrete-event systems using queues and servers. With SimEvents software you can create a discrete-event simulation model to simulate the passing of entities through a network of queues, servers, gates, and switches based on events. The software provides an integrated environment for modeling hybrid dynamic systems containing continuous-time, discrete-time, and discrete-event components.

A key concept that SimEvents software adds to the Simulink environment is that of *entities*, which are discrete items of interest in a discrete-event simulation. For example, entities could represent messages to be communicated or parts to be assembled. Entities can carry data in one or more scalar structures called *attributes*. For example, attributes could represent destinations of messages or dimensions of parts.

The libraries in SimEvents software contain blocks that can

- Create entities
- Store entities in a queue
- Serve or delay entities
- Forbid or allow entities to depart, depending on specified criteria
- Manipulate the paths on which entities travel

- Attach data or timers to entities
- Create plots using data from entities or statistics gathered during simulation
- Manipulate or generate discrete events that can affect the behavior of blocks and entities
- Control the simulation timing in situations where event-driven behavior and time-driven behavior interact

Compatibility Summary for SimEvents Software

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided in the description of the new feature or change.

Version (Release)	New Features and Changes with Version Compatibility Impact
Latest Version V3.1 (R2010a)	See the Compatibility Considerations subheading for each of these new features or changes: <ul style="list-style-type: none"> • “Block Operations Information in Debugger” on page 5 • “Changes in Behavior of Pending Entity Signals” on page 5 • “Parameters and Parameter Values Being Removed” on page 10
V3.0 (R2009b)	None
V2.4 (R2009a)	See the Compatibility Considerations subheading for each of these new features or changes: <ul style="list-style-type: none"> • “Event Logging Options Removed from Configuration Parameters Dialog Box” on page 15 • “Discrete Event Signal to Workspace Block Clarifies Timing” on page 17
V2.3 (R2008b)	None
V2.2 (R2008a)	None

Version (Release)	New Features and Changes with Version Compatibility Impact
V2.1 (R2007b)	<p>See the Compatibility Considerations subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> • “Processing Events Via the Event Calendar Instead of Immediately” on page 29 • “Enhanced Support for Multiple Simultaneous Transitions in Switches and Gate” on page 31
V2.0 (R2007a)	<p>See the Compatibility Considerations subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> • “Attributes Support Multidimensional Values” on page 34 • “Timeout Feature Establishes Entity Time Limits” on page 35 • “Output Switch Block Options for Storage and Initial Condition” on page 37

Version (Release)	New Features and Changes with Version Compatibility Impact
V1.2 (R2006b)	<p>See the Compatibility Considerations subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> • “Event Translation Block Supports Delay from a Signal” on page 44 • “Routing Blocks Support Unlimited Entity Ports” on page 45 • “Initial Outputs of SimEvents Blocks” on page 46 • “History Options and Other Changes in Scope Blocks” on page 48 • “Parameters for Lognormal Distribution” on page 49 • “Livelock Detection” on page 50
V1.1 (R2006a)	<p>See the Compatibility Considerations subheading under “Replicate Block Supports Partial Replication” on page 51.</p>
V1.0 (R14SP3+)	<p>Not applicable</p>