JaamSim

Release Notes

Outstanding Feature Requests

Feature	Description	Difficulty	Importance
Inputs in a separate file	Allow the TimeSeries, EventSchedule, and ValueSequence objects to accept input data in a separate file.	Medium	Medium
State for delayed entities	Provide the option of changing the state of an entity whose processing by a Server, etc. is delayed by a threshold closure or by a maintenance or breakdown event.	Low	Low
File Writing	Add new object that can write custom text to an output file.	Low	Medium
EntityDelay Thresholds	Add Thresholds inputs to EntityDelay	Medium	Medium
RelativeEntity for end of an EntityConveyor	Provide the ability to position an entity relative to the end of an EntityConveyor instead of its start.	Medium	Low
Preempt	Provide the ability to preempt a Server by a higher priority entity that arrives in its queue.	Medium	Low

Change Log

Changes in release 2026-08:

- **Bug fix for drag-and-drop to a SubModel**. A bug that prevented objects from being dragged and dropped to a SubModel region has been fixed. The bug was introduced accidentally in release 2026-03.

Changes in release 2026-07:

- **Java version**. JaamSim now runs correctly under the latest Java versions (Java 26 and 27). However, support for older versions prior to Java 17 have been dropped.

Changes in release 2026-06:

- **Window positions**. The positions of the various JaamSim windows have been adjusted to prevent gaps when running under Windows 10 and 11.
- **Example models**. The list of example models under Help > Examples are now sorted correctly into alphabetical order.
- **Vector inputs**. The inputs 'Position', 'Alignment', 'Size', and 'Orientation' now accept expressions.

Changes in release 2026-05:

- **Redo/Repeat button**. The Redo button has been modified to include the 'repeat' function using the same conventions at Microsoft Word and Excel. When a new object is selected, the repeat button executes the last command for the new object, if it is applicable. Keyboard shortcuts for 'repeat' are Ctrl+Y and F4.
- **'FractionCompleted' output for EntityConveyor**. This output returns the fractional distance travelled by each entity on the conveyor.
- **'TriggerPointList' input for EntityConveyor**. This input specifies one or more fractional distances along the conveyor at which the conveyor will notify any object that includes the conveyor in its 'WatchList' input. Any entity that reaches a trigger point will cause the watchers to be notified.
- **'EntityConveyor Example – with TriggerPointList' model**. The example model demonstrates the use of the TriggerPointList input.
- **Example models re-organised into a tree structure**. This example models have been re-organised into a tree structure with the same headings as the Model Builder, e.g. Graphics Objects, Probability Distributions, etc.

Changes in release 2026-04:

- **'ServiceDuration', 'ServicePerformed', and 'FractionCompleted' outputs for EntityProcessor**. These outputs are provided to make the EntityProcessor object consistent with the Server object. Note, however, that for an EntityProcessor these outputs return an array of values (one for each entity being processed) instead of a single value.
- **Option to delete an unused DisplayModel**. Two objects are created when an image file or 3d file is imported: 1) a DisplayModel that holds the graphics data, and 2) a DisplayEntity that appears in the model. The DisplayModel is an ImageModel in the case image file and a ColladaModel in the case of a 3d file. If the imported DisplayEntity is deleted, it is easy to forget to delete the DisplayModel that references the file. To avoid a model becoming cluttered with unused DisplayModels, the user is now prompted to delete a DisplayModel that becomes unused when the only DisplayEntity that references it is deleted.

- **'ScriptedViewDirection' input for View**. This input replaces the 'ScriptedViewCenter' input to be consistent with the 'ViewDirection' input that was introduced in release 2026-03.
- **'UserList' output for DisplayModel**. This output lists all the objects that use this DisplayModel for their graphical display.
- **Bug fix for model execution on multiple threads**. The ability to run models on multiple threads has been restored. This feature was broken accidentally in release 2026-01.
- **Bug fix for file streams**. Connections to input and output files are closed correctly when an error occurs.

Changes in release 2026-03:

- **Point-of-Interest**. Zooming and orbiting actions with the mouse are centred on the point-of-interest in 3d space. The rules for setting this location have been refined to improve user experience. It is now set when an object is dragged-and-dropped, when it is dragged to a new position, and when it is right- or left-clicked. The point-of-interest is unchanged by a right- or left-click when no object is under the cursor. However, when the coordinate grid is visible, the 'XY-Grid' object provides an object on the xy-plane to be detected by a mouse click. Formerly, the point-of-interest would be set by default to the point on the xy-plane that was under the cursor when no object was present. For 3d models located far from the xy-plane, this behaviour could cause large unwanted movements of the view camera for zooming and orbiting which required the user to click the undo button to restore the view.
- **'XY-Grid' object**. This existing object, which displays the coordinate grid on the xy-plane, has been changed to use a ShapeModel instead of a ColladaModel to provide its graphics. A new 'GRID' option has been added to the 'Shape' input for ShapeModel for this purpose. A new feature is that the background colour and transparency for the XY-Grid object can be set using the FillColour input for the 'Grid100x100' ShapeModel. This input is set to 'white 0', indicating a white colour with 100% transparency so that it is not visible, but is detected by a mouse click. Models created with earlier versions of JaamSim can still be loaded into this version.
- **'ViewDirection' input**. This input for the View object sets the direction of its camera. Formerly, the 'ViewCenter' input was used for this purpose. Models created with earlier versions of JaamSim can still be loaded into this version.
- **Astronomical distances**. The 3d graphics code has been refined to work correctly for the enormous distances required by astronomical models. Objects separated by a distance of up to 100 billion light-years can be displayed together.
- **Internal improvements**. The code for view cameras and rendering has been streamlined.

Changes in release 2026-02:

- **Optimisations for Java 25**. The software has been optimised for use with the Java 25 virtual machine. Some users may see a performance improvement. The software is still backwards compatible with the Java 8 virtual machine.
- **Internal improvements**. Cleanups have been performed to simplify and improve window handling.

Changes in release 2026-01:

- **Outputs for Model Inputs**. All model inputs now have a corresponding output that displays its current value and that can be used in an expression. Previously, only a limited number of inputs had a corresponding output, and in some cases, they did not work correctly in an expression.
- **'Parent' input for all objects**. The Parent input allows the user to designate an object to be the child of another object. When an object is cloned or copied, all its child objects are clone or copied along with the

parent. One use for this feature is to display additional information for the clones created by an EntityGenerator.

- **'Set Parent and RelativeEntity' menu item**. The context menu (right-click) now includes an entry for setting the Parent and RelativeEntity inputs. In most cases, the position for a child entity should be relative to its parent.
- **Drag-and-drop for all objects**. Any object that has a child entity or a clone will appear in the Model Builder under the heading 'User-Defined Objects'. SubModels are automatically included by this condition. When dragged and dropped, a user-defined object creates a clone..
- **'Paste Clone' menu item**. The context menu now includes an entry for 'Paste Clone' that will insert a clone of a copied object. By creating a clone, the original object will be added automatically to the Model Builder under 'User-Defined Objects', making it easier to create additional clones.
- **'SimShape' object added to Process Flow library**. The SimShape object is a 2D version of SimEntity that is based on the Shape object and includes additional inputs for Filled, FillColour, Outlined, LineColour, and LineWidth.
- **'Job Shop Example' model**. This example model under Help > Examples is based on the problem described in Chapter 2.7 of "Simulation Modeling and Analysis", 5th Edition, by Averill M. Law. The model demonstrates the use of SubModel, SimShape, and child entities.
- **'Image' object added to Graphical Objects library**. This object has been added to make it easier for users to learn how to add images to their models. Formerly, it was necessary to use a DisplayEntity for this purpose.
- **'PixelSize' output for ImageModel**. This output returns the width and height of the image in pixels.
- **'AlignEntities' input for EntityConveyor**. The default setting of FALSE for this input allows an accumulating conveyor to use the same positioning of conveyed entities as a non-accumulating conveyor. It should be set to TRUE for an accumulating conveyor that conveys entities with different lengths.
- **'EntityConveyor Example - with Entities of Different Length' model**. This example model under Help > Examples shows how to use an accumulating conveyor when the conveyed entities that have different lengths.
- **'WatchList' input for EntityConveyor**. This input allows a change to the TravelTime input to be recognised by an EntityConveyor when one of the objects in the list changes state. Otherwise, a change in the TravelTime input is recognised by the arrival or exit of an entity. A WatchList entry should be provided whenever an expression is entered to the TravelTime input. Otherwise, the TravelTime expression will tested at each event time (i.e. a conditional event) to determine whether or not its present value has changed.
- **'PresentTravelTime' output for EntityConveyor**. This output returns the travel time that is being used at present for the conveyor. This time can differ from the value for the TravelTime input if the conveyor has not been updated by the arrival or exit of an entity, or been triggered by an entry in its WatchList.
- **'ServerAndQueue' object removed from the Model Builder**. This object is no longer available for dragging and dropping into a model. For backwards compatibility, it is still available for use in existing models.
- **Updated 'Nested SubModel Example' model**. This example model under Help > Examples has been updated to use a SubModel instead of the ServerAndQueue object, which is no longer available.
- **'RelativeEntity' input accepts an expression**. The RelativeEntity input can now accept an expression that returns an entity.
- **'Name' and 'Parent' inputs locked for Simulation**. It is not possible to change the name or parent for the Simulation object.

- **Default 'Parent' input set to null**. Previously, the default Parent was the Simulation object.
- **'NumberDestroyed' output for EntitySink**. Returns the total number of entities that have been destroyed by the EntitySink including those during the initialisation period (warm-up). This output is analogous to the 'NumberGenerated' output for EntityGenerator. These two outputs allow the number of entities in progress (i.e. the work in progress) to be calculated.
- **Speedup for EntityProcessor with variable capacity**. If its Capacity input is an expression, an EntityProcessor checks whether its present capacity value has changed at each event time (i.e. a conditional event). However, if a WatchList input is provided, the capacity is now checked only when triggered by a state change to the objects in the WatchList, which is much more efficient.
- **Expression format used for numerical outputs**. All numerical outputs are now displayed in expression format to help make it clear that they can be used in expressions.
- **Bug fix for 'Show Labels'**. An error message is generated if the user attempts delete an EntityLabel while the 'Show Labels' button is turned on. Formerly, the label would be deleted successfully, but it would be restored by the 'Show Labels' button if the model was restarted or if it was saved and reloaded.
- **Bug fix for Attributes**. Attribute values are returned with the correct unit type prior to the start of a model. Formerly, the unit type was unassigned after a model had been loaded but not started.
- **Bug fix for copying a SubModel**. Fixes a bug that occurred when a SubModel was copied with the 'Show Labels' button turned off.
- **Bug fix for a SubModel label**. Fixes a bug that caused the EntityLabel for a SubModel to disappear when the 'Show SubModels' button was turned off.
- **Bug fix for deleting a SubModel**. A SubModel with clones can now be deleted and restored successfully.

Changes in release 2025-10:

- **'EntityDependentList' output**. This output returns a list of the objects whose inputs reference the present object.
- **Bug fix for 'EntityReferenceList' output**. This output now includes any objects that are included in the 'AttributeDefinitionList' input.
- **Bug fix for 'NumberRequired' input**. This input for the Assemble and Combine objects can now accept expressions that use the 'obj' output.
- **Bug fix for window size**. A bug has been fixed that caused the Control Panel window to be occasionally shown as a small square when the software is first launched.
- **Bug fix for View window position**. A bug has been fixed that frequently caused the View window to be displaced slightly from its normal position when the software is first launched.
- **Bug fix for error handling**. A second stack trace has been removed from the software output that was caused by a null-pointer exception that occurred during the handling of a runtime error.

Changes in release 2025-09:

- **Internal changes**. Changes have been made to the JaamSim code to resolve some software design issues. These should not affect the user experience.

Changes in release 2025-08:

- **'FractionCompleted' output for Process Flow objects**. In addition to Server and EntityGenerator, which already have this output, the 'FractionCompleted' output is now available for: Assemble, Combine, AddTo, Pack, RemoveFrom, Unpack, and EntityGate.

- **'simTimeForDate' function**. This function calculates the simulation time that corresponds to a specified calendar date.
- **'Monthly Totals Example' model**. This example model under Help > Examples shows how the 'simTimeForDate' function can be used to collect operating statistics at the start of each calendar month.
- **JOGL 2.6.0**. The JOGL software that provides the OpenGL support for 3d graphics in JaamSim has been updated to the latest release. It should have no effect on JaamSim. Please report any anomalous behaviour to the JaamSim forum.
- **Bug fix for XYGraph**. A bar graph created using the XYGraph object can now display a single bar. Previously, nothing was shown until there were two or more bars to display.
- **Bug fix for Run Progress**. The Run Progress display is now shown with a progress bar for each thread as soon as the Play button is pressed. Previously, due to a bug introduced accidentally in release 2025-06, the progress bars for each thread were shown only after all threads had been started.

Changes in release 2025-07:

- **Increased range for 'Speed Multiplier' input**. The upper limit for the 'Speed Multiplier' input has been increased from 1 million to 100 million. The new limit is equivalent to 3.2 years of simulated time per second of real time. To increase readability, the display now includes comma separators for thousands, millions, etc.
- **Bug fix for EntityConveyor**. A non-accumulating EntityConveyor now displays the entities being transported in their correct positions when the conveyor is included in a sub-model or whose 'RelativeEntity' input has been set. A bug introduced in release 2025-05 caused the entities to be displaced from their correct positions.
- **Bug fix for auto-complete menu**. The entries in the auto-complete menu for outputs in the Input Builder are now shown in the correct order. Previously, the entries for attributes and custom outputs were displayed in order of definition.
- **Bug fix for Input Editor**. The scroll position for the Input Editor is now retained when a new object is selected for editing. Previously, the highlighted keyword would be scrolled to the top of the editor.
- **Bug fix for 'AttributeDefinitionList' input**. An 'AttributeDefinitionList' input that is set before the 'RecordEdits' in the configuration file is now handled correctly. A bug introduced in release 2025-05 caused this input to become corrupted if the model was saved after any other input for the object in question was edited.
- **Bug fix for window title**. The default window title is now "Model1 - JaamSim". A bug introduced in release 2025-06 set the title to a blank string.
- **Bug fix for 'Remaining Time' and related displays**. The displays for 'Remaining Time', 'Run Progress', and 'Achieved Speed Multiplier' are now retained when the simulation run is paused. A bug introduced in release 2025-06 caused them to be cleared.

Changes in release 2025-06:

- **Bug fix for 'AttributeDefinitionList' input**. The expression that is used to set the initial value for an attribute can now reference the attributes and/or custom outputs for other objects. Previously, attributes could be defined in this way, but the saved configuration file would not load correctly. Furthermore, execution on multiple threads would cause similar problems.
- **Bug fix for window sizes**. Tool and view window sizes are set more reliably to their default values on startup. On occasion, there would sometimes be gaps between windows on launching JaamSim or the software would launch with the windows minimized.

- **Bug fix for Big Lean Library examples under Java 8**. A bug has been corrected that prevented the Big Lean Simulation Library examples from working correctly under Java 8 when accessed through Help > Examples.

Changes in release 2025-05:

- **Big Lean Simulation Library**. The simulation models for lean manufacturing created by Jerry Levasseur have been added as example models under Help > Examples. The accompanying videos have been added to the JaamSim YouTube channel for easy access.
- **Example model files**. Some of the example models under Help > Examples require an additional input file to be processed. The example models for FileToVector, FileToMatrix, and FileToHashMap are the ones most commonly encountered. When one of these models is opened by the user and then saved as a local copy, the user is prompted to save a local copy of any additional files.
- **Auto-complete menus for Input Builder**. The auto-complete menu for functions has been modified to ignore the space bar. For the other two auto-complete menus (outputs and objects), pressing the space bar accepts the present menu entry. This change was made to allow the user to continue typing when the auto-complete menu for functions is not needed, which is often the case.
- **Bug fix for '2D View' button**. This button is now disabled when the 'Movable' input for the present view has been set to FALSE.
- **Bug fix for EntityConveyor**. When accumulating conveyors were introduced in release 2023-02, the positions of the entities on the conveyor were changed to align their left-side instead of their centre. This change was needed for the spacing between entities to be correct. However, it causes non-accumulating conveyors to have some undesirable consequences, particularly when the conveyor travels in the y-direction. For this reason, non-accumulating conveyors have been changed back to their original logic.
- **Bug fix for SubModel**. A bug has been fixed that caused the components of a SubModel clone to inherit the wrong value for its Region input. This error was introduced accidentally in release 2024-08.
- **Bug fix for 'CloneList' output**. When a model was executed in headless mode, the CloneList output failed to include any clones created by an EntityGenerator.
- **Bug fix for Images**. An Image now appears correctly when its 'Orientation' input rotates it out of the xy-plane. Previously, an image would become increasingly dark as the angle from the xy-plane was increased. It became completely black when the image was vertical. This bug affected the appearance of any model icon that was rotated out of the xy-plane.

Changes in release 2025-04:

- **Bug fix for Event Processing**. A bug has been fixed that caused some models to freeze in the middle of a run, with the error message "Spurious wakeup in EventManager eventStack.Thread[processthread-1,6,main]" printed to System.out. The bug was introduced accidentally in release 2025-02.

Changes in release 2025-03:

- **Bug fix for Real Time**. A bug has been fixed that caused some models to run excessively slowly when the 'Real Time' button was selected. The bug was introduced accidentally in release 2025-02.

Changes in release 2025-02:

- **Event Viewer cleanup**. When a model is started, the Event Viewer now shows only the 'startUp' events for the objects that have been defined by the model. Formerly, a very large number of unnecessary

startUp events would be executed for pre-defined objects such as Units and DisplayModels that do not need to be started in this way.

- **EntityProcessor statistics**. The EntityProcessor object now has a complete set of outputs such as "UnitsInUseAverage" that are analogous to those for the Resource object.
- **Error checking for AddTo, RemoveFrom, and Unpack**. These objects now check that an EntityContainer or related object is received from the 'WaitQueue' for the RemoveFrom and Unpack objects, and from the 'ContainerQueue' for the AddTo object. Formerly, this type of error would result in an unhelpful Java error message.

Changes in release 2025-01:

- **Auto-complete menu in Input Builder**. An auto-complete menu is cleared when a user presses the Escape key or clicks anywhere in the input field.
- **"PRESENT VALUE" display in Input Builder**. For the 'CustomOutputList' and 'AttributeDefinitionList' inputs, the PRESENT VALUE display now shows the present value for each of the custom outputs and attributes that were defined.
- **ExpressionStatistics inputs and outputs**. The 'TargetPercentiles' input and a number of outputs were added to ExpressionStatistics to make it consistent with the Statistics object.
- **'Cursor Position' popup.** The popup description for the Cursor Position display has been corrected to refer to a position in 3D space.
- **Custom Output Report.** The blank line between scenarios has been removed when only one replication is performed.
- **Bug fix for View windows**. A bug has been fixed that sometimes caused a view window to rotate randomly by 90 or 180 degrees while it is dragged in the x-y plane. This bug first appeared in release 2024-04.
- **Output Viewer displays error message**. When a custom output cannot be evaluated, the relevant error message is now displayed in the Output Viewer. Formerly, the text "Cannot evaluate" was displayed.
- **Improve error message for cfg file problems**. A common error in a cfg file that has been edited manually is to forget to include a closing brace ('}') or a closing single quote mark ('). When this occurs, the input parser reads to the end of the file and prints an error message that appears at the end of the .log file. The error message now prints the first line in file that could not be parsed, which makes it much easier to locate the missing closing brace or single quote mark.

Changes in release 2024-09:

- **Auto-complete feature for Input Builder**. A new auto-complete feature for function names and special names such as 'this', 'simTime', 'sub', etc. have been added to the Input Builder. This feature complements the auto-complete features for entity names and output names.
- **Documentation pop-ups for Input Builder**. Documentation pop-ups now appear in the Input Builder when the mouse is positioned over each part of an expression. For example, if the mouse is positioned over the 'size' function, then a pop-up appears that describes this function, including its inputs, outputs, and examples of its use.
- **Functions 'discrete' and 'continuous'**. These functions randomly select from user-defined discrete and continuous distributions respectively. They are analogous to the DiscreteDistribution and ContinuousDistribution objects in the Probability Distributions library.

Changes in release 2024-08:

- **Resources seized by a DowntimeEntity**. Inputs 'ResourceList' and 'NumberOfUnits' have been added to DowntimeEntity to specify the resources that must be seized before the 'repair' activity for the downtime event can begin. The resources are released when the repair activity is finished and the DowntimeEntity resumes its 'Working' state. If the specified resources are not available to be seized, then the DowntimeEntity is set to the state 'WaitingForResources' and work is halted. This feature could be used, for example, to model a limited number of maintenance workers that perform the planned and unplanned maintenance for a number of processes. An example model 'DowntimeEntity Example - with ResourceList' has been added under Help > Examples to illustrate the use of this feature.
- **Entity reference arrows**. When the 'Show References' button is selected, the reference arrows now point from the centre of the first object to centre of the second object. Previously, arrows pointed to/from the 'Position' input values, which is not always the centre of each object. Furthermore, a small dot has been added at the beginning of each arrow.
- **Undo for SubModel deletion**. A single entry is now shown in the 'Undo' button's list when a SubModel is deleted. Formerly, each of the SubModel's components had a separate entry in the list.
- **Bug fix for output reports**. The 'RunDuration' and 'InitializationDuration' outputs are now recorded in the .rep and .log reports in the same way that they appeared with releases of the software prior to 2024-07.
- **Bug fixes for EntityLabel**. The EntityLabel is now hidden for an entity that has exited the model and been added to a pool for re-use.
- **Bug fix for selected entity**. The entity selected for display in the Input Editor and Output Viewer is cleared when the entity exits the model.

Changes in release 2024-07:

- **EntityLabel recast as a child of its target object**. Instead of naming the EntityLabel for an object by appending "_Label" to the object's name, it is now simply a child of the object. For example, the EntityLabel for the object [Server1] is now the child object [Server1.Label] instead of being a separate object named [Server1_Label].
- **Bug fix for line width**. Lines with a width of zero pixels are now shown as invisible. Formerly, most computers displayed a line with zero width in the same way as one with a width of 1 pixel. However, on some computers, a zero-width line would cause the view windows to stop working.
- **Bug fix for EntityLabel positions and formats in a SubModel clone**. The position and format of an EntityLabel in a SubModel clone are now synchronized with the position and size of the EntityLabel in the prototype SubModel.
- **Bug fix for model inputs as outputs**. Name collisions between existing outputs and the new ones that echo an input value have been resolved.
- **Bug fix for Object Selector**. The Object Selector is updated correctly when a generated entity is re-cycled by an EntityGenerator.
- **Bug fix for EntityLabel**. Entity labels are shown correctly when a new model is opened.
- **Bug fix for saved and imported SubModels**. A SubModel is now saved in such a way to avoid errors involving its Region.
- **Bug fix for window positions**. A bug has been fixed that occasionally caused the tool and view windows to be incorrectly sized and positioned when a model is opened.

Changes in release 2024-06:

- **Model inputs as outputs**. Most of the inputs to each simulation object are now available as outputs, and can be used in expressions. These new outputs appear at the bottom of the Output Viewer.
- **Inputs to SubModel clones**. All inputs to the components of a SubModel clone can now be overridden in the Input Editor. Formerly, inputs that referenced the parent SubModel were show in grey and could not be overridden.
- **EntityLabel shows the local name**. To reduce clutter, an EntityLabel now shows its entity's local name instead of its global name. For example, the global name of Server1 contained in SubModel1 is 'SubModel1.Server1', whereas its local name is 'Server1'.
- **'Name' input**. All objects now have a 'Name' keyword which can be used to change the object's local name. This method is more convenient for renaming an object than using the Object Selector or the EntityLabel. Note that only the local name can be changed for an object.
- **Control Panel size**. The main window for JaamSim is now fixed in size and its 'maximize' button has been disabled.
- **'Lock Windows' button**. This new button in the button bar allows the user to prevent the tool and view windows from being re-positioned or re-sized accidentally.
- **'NumberStarted' and 'NumberCompleted' outputs**. These new outputs for DowntimeEntity return the number of downtime events that have been started and completed up to the present time.
- **'Children' output**. This output lists the objects that are subordinate to the present object. For the case of a SubModel, these objects are its components.
- **Bug fix for window sizes**. A bug has been corrected that prevented the position and size of the tool windows from being preserved between the saving and relaunching of JaamSim. The bug was introduced accidentally in release 2020-03.

Changes in release 2024-05:

- **Pixel scaling**. High resolution monitors that use pixel scaling are now supported natively. The 'Display Menu' app is no longer required for Mac computers with a Retina monitor.
- **Improvements to the Finder tool**. A list of objects copied to the Finder now generates a menu entry for each object in the list.
- **Context menus for Input Editor and Output Viewer**. These windows now provide a context menu when the selected row is right-clicked. For the Input Editor, the options are Cut, Copy, Paste, Delete, and Find. For the Output Viewer, the options are Copy and Find.
- **Faster startup for complex models**. Models with many SubModels that contain large numbers of objects now start executing after only a short delay. Formerly, these models could take several minutes before simulated time began to advance.
- **Improved error messages**. Minor improvements to the formatting of error messages.
- **Bug fix to the use of 'this' in expressions**. This bug affected expression inputs that included 'this' for ExpressionEntity, InputValue, Polynomial, SineWave, SquareWave, and WeightedSum.

Changes in release 2024-04:

- **Snap-grid for pan action**. When view is panned with the mouse (left-drag), the view moves in increments of the 'Snap Grid Spacing' input in the same way as an individual object that is dragged.
- **Cursor Position**. This display now shows the x, y, z coordinates on the surface of the first object under the mouse pointer. Formerly, it showed the position on the x-y plane under the mouse pointer.

- **Simplified jar file launching under Java 17**. The JaamSim jar file can be launched under Java 17 and above using the standard command line, i.e. `java -jar JaamSim2024-04.jar`. Formerly, it was necessary to a much longer command line, i.e.
  `java --add-exports java.desktop/sun.awt=ALL-UNNAMED -jar JaamSim2024-03.jar`.
- **SubModel file with external references**. A SubModel that includes external references can now be saved. A warning message will be shown instead of an error if the SubModel includes any external references.
- **Bug fix for EntityLogger**. All generated entities are now initialized correctly so that they do not trigger spurious state-change records for an EntityLogger when its TraceEntityStates option is used.
- **Bug fix for EntitySink**. The 'obj' output for an EntitySink now returns the entity that it received most recently. Formerly, it always returned null.
- **Bug fix for zoom action**. Prevents an unexpectedly large zoom action from the mouse wheel when a View window is first created.
- **Bug fix for pan action**. A view can now be panned correctly using the mouse when the view has been zoomed-in to an extremely small scale. Formerly, panning stopped working under this condition.
- **Bug fix for EntitySystem**. The state of an EntitySystem is now updated correctly when one of the objects in its WatchList changes state from 'Idle' to 'Maintenance'.
- **Bug fix for example models**. The models found under Help > Examples have the PrintReport input set to FALSE.

Changes in release 2024-03:

- **Bug fix for input files**. Object definitions that depend on other objects such as clones and child objects are now saved to the input file in the correct sequence. Formerly, it was possible receive input error messages because the clone of an object was defined prior to its prototype, or the child of an object was defined prior to its parent.
- **Bug fix for expression formatting**. Carriage returns that are entered in the Input Builder to format a complex expression are now retained correctly. Formerly, a bug introduced in release 2023-08 caused the carriage returns to be deleted when the Accept button was clicked.
- **Bug fix for Text object**. Text objects now handle a change to their UnitType inputs correctly. Previously, a change to this input would be recorded but it would not change the internal logic.
- **Bug fix for Object Selector**. When the Object Selector has a horizontal scroll bar, the bar is now unaffected while the model is running. Formerly, a bug caused the scroll bar to constantly reset itself the left while the model was running.

Changes in release 2024-02:

- **GLTF and GLB format for 3D objects**. 3D objects in GLTF and GLB format can now be imported in the same way as other 3D formats such as OBJ and DAE. These new formats are becoming the industry standard for 3D content and are much more powerful than the older OBJ and DAE formats. Animated 3D objects can be created in the free open-source software Blender, exported in either of these formats, and then imported into JaamSim. At present, only rigid-body type animations are supported by JaamSim. This type of animation is suitable for typical engineering subjects such as manufacturing equipment, vehicles, etc.
- **'Durations' output for ColladaModel**. This output returns a list containing the duration of each animation provided by the 3D model.
- **Bug fix for Combine and Assemble**. A bug has been fixed that caused some of the entities in a model containing these objects to be invisible.

- **Bug fix for nested SubModels**. A SubModel that contains a clone of another SubModel can now be dragged and dropped successfully. Previously, an error would occur that prevented the drag and drop process from being completed.
- **Bug fix for SubModels containing a ExpressionThreshold**. An error has been corrected that under some circumstances could prevent a SubModel containing an ExpressionThreshold from being dragged and dropped successfully.

Changes in release 2024-01:

- **XYGraph object**. This object allows line and bar graphs to be displayed for any set of x-y points. Its primary use is to display the histograms recorded by Statistics, Queue, and Resource objects.
- **'Statistics and XYGraph Example'**. This new example model under Help > Examples demonstrates the use of the new XYGraph to display a histogram of waiting times collected by a Statistics object.
- **Cosmetic changes to Graph**. The inputs LineColours, LineWidths, SecondaryLineColours, and SecondaryLineWidths have been moved to a new 'Format' tab in the Input Editor. New default values in hours have been set for the inputs in the 'X-Axis' tab.
- **'QueueLengthFractions' and 'QueueLengthCumulativeFractions' outputs**. These new outputs for Queue return the fraction of time and cumulative fraction of time that the number of entities in the queue is 0, 1, 2, etc.
- **'UnitsInUseFractions' and 'CumulativeUnitsInUseFractions' outputs**. These new outputs for Resource, ResourcePool, and ResourceUnit return the faction of time and the cumulative fraction of time that the number of resource units in use is 0, 1, 2, etc.
- **'HistogramBinCumulativeFractions' output for Statistics**. This output returns the cumulative distribution for the existing 'HistogramBinFractions' output.
- **'HistorgramBinUpperLimits' output for Statistics**. This output returns the value corresponding to the right-hand side of each histogram bin. It is intended to be used with the 'HistogramBinCumulativeFractions' in the inputs to an XYGraph that displays the cumulative distribution for a histogram.
- **'TargetPercentiles' input and 'PercentileValues' output** for Statistics. This input and output allow the user to specify one or more percentiles values to be estimated from the histogram results. For example, entering `90 95` to the TargetPercentiles inputs would cause the PercentileValues output to return the values that are larger than 90% and 95% of the recorded sample values, respectively.
- **'StandardDeviationOfTheMean' output for Statistics**. This output has been deleted because it is only valid when the recorded values are normally distributed. This is rarely the case for the statistics collected from a single simulation run, which made this output misleading.
- **Automatic updating for SubModel**. When a new object is dragged and dropped to a SubModel, any clones of this SubModel are updated automatically to include a clone of the new object.
- **Input Conditioning**. Single quotation marks are now added automatically to inputs that accept a list of expressions when they are entered in the Input Editor or Input Builder.
- **'Location' input for distributions**. This input has been added to BetaDistribution, LogLogisticDistribution, and LogNormalDistribution.
- **Bug fix for LogLogisticDistribution**. This distribution now accepts a Shape input that is less than or equal to 2.
- **Bug fix for View windows**. The camera position controls now work correctly when the initial camera angle is parallel to the x-y plane.

- **Bug fix for Object Selector**. Right-clicking on an object the Object Selector now generates the context menu more reliably. Formerly, the context menu appeared only about 50% of the time.
- **Bug fix for ExpressionLogger**. A log entry scheduled for the end of a simulation run is now recorded correctly. A bug introduced in release 2023-09 had caused this entry to be omitted.
- **Bug fix for LogNormalDistribution**. The 'CalculatedStandardDeviation' output now returns the correct value when the 'Scale' input is set.
- **Bug fix for multiple threads**. Bugs related to the SubModel object and the 'RandomSeed' input have been corrected that could prevent a model from executing on multiple threads.

Changes in release 2023-09:

- **ExpressionStatistics Object**. This new object can be used to determine the time-based average and standard deviation of the numerical value returned by a specified expression. Histogram data for the value can also be generated by the object.
- **Bug fix for ExpressionLogger**. Multiple entries at the same simulation time have been eliminated in the log files generated by the ExpressionLogger object. Previously, duplicate entries could appear when triggered by separate entries in the ValueTraceList and StateTraceList inputs, or when an object changes state two or more times at the same simulation time, e.g. from 'Busy' to 'Idle' and back to 'Busy'.

Changes in release 2023-08:

- **Bug fix for Input Builder**. An error introduced accidentally in release 2023-07 has been corrected. The error prevented the Input Builder from being used with inputs such as RunOutputList, AttributeDefinitionList, AttributeAssignmentList, and many others.
- **Input conditioning**. Single quotation marks are now added automatically to the following additional input types: AttributeDefinitionList, AttributeAssignmentList, and CustomOutputList. If required, the single quotes are added when these inputs are entered in the Input Editor or Input Builder.

Changes in release 2023-07:

- **Support for Mac computers with M1, M2, etc. CPU chips**. JaamSim now works correctly on Mac computers with the M1, M2, etc. CPU chips as well as those with the older Intel chips. Formerly, Mac computers with the M1, M2, etc. chips could not display the 3d graphics in the View windows.
- **'Recalculate' input for Pack and AddTo**. When set to TRUE, this input instructs the Pack and AddTo objects to recalculate the values for their 'NumberOfEntities' and 'NumberToStart' inputs each time an entity arrives to the wait queue until such time as packing can begin. Otherwise, these inputs are assigned fixed values when they are first evaluated for each EntityContainer.
- **Default value for 'EndingScenarioNumber' input**. This input now defaults to the value for the 'StartingScenarioNumber' input. Formerly, it defaulted to 1. This change avoids a potential input error by ensuring that at least one scenario will be executed.
- **Entity-list inputs accept an expression**. Many inputs that accept a list of entities can now accept a list of expressions that return entities.
- **Bug fix for View window**. A model that is launched from the command line using the name of its configuration file now displays the View window correctly. Formerly, the window would be displayed at the top left of the screen with just the title bar showing.
- **Windows 32-bit executable discontinued**. We have upgraded the internal 3D library to be able to support newer M1 and M2 Macs, but we have had to retire the 32-bit Windows version as it has been dropped upstream.

Changes in release 2023-06:

- **Colour inputs accept an expression**. All colour inputs now accept an expression that returns a colour name or an array of RGB values.
- **Boolean inputs accept an expression**. All boolean inputs now accept an expression that returns 1 (TRUE) or 0 (FALSE).
- **Numerical inputs accept an expression**. All remaining inputs that accept a single number with or without units, now accept an expression that returns such a number.
- **Up and down keys enabled for dropdown menus**. The up and down arrow keys now work correctly with the dropdown menus displayed by the 'DisplayModel', 'Font', 'Font Colour', 'Line Colour', and 'Fill Colour' buttons, including their preview feature. Formerly, it was not possible to use the arrow keys with these menus after scrolling with the mouse. Furthermore, the preview feature did not work with the arrow keys, which limited their usefulness.
- **Preview enabled for colour menus in the Input Editor**. The dropdown menu for colour inputs, such as the 'FillColour' keyword, now display a preview as each menu item is highlighted with either the mouse or an arrow key.
- **Improved description for 'RecordEntityStateTimes' input**. The description for this input to the Statistics object now makes it clear where to find the statistics for the recorded entities.
- **'PolylineWidth' input for Fluid objects**. The FluidPipe and FluidFixedFlow objects now accept an input for PolylineWidth, similar to other polyline-type objects such as Polyline, EntityDelay, and EntityConveyor.
- **Bug fix for FluidTank**. The initial content of a FluidTank is now displayed as soon as the 'InitialVolume' input is set.
- **Bug fix for 'RotateEntities' input**. The entities moved by an EntityConveyor or EntityDelay object are now rotated only when the 'RotateEntities' input is set to TRUE.
- **Bug fix for 'MatchValue' output**. The value for this output is updated whenever a Pack or AddTo object checks its queue. Formerly, the value was set only when a Match value had been chosen for the present EntityContainer and packing had begun.
- **Bug fix for BarGauge2D**. A BarGauge graphic is now displayed correctly for any object that uses the BarGauge2D for its DisplayModel setting. Formerly, only the BarGauge object would display the BarGauge2D DisplayModel. All other objects simply disappeared when this setting was used.
- **Bug fix for Copy and Paste**. The various Copy and Paste functions in the user interface now work in exactly the same way as Ctrl-C and Ctrl-V. These functions include: context menu entries (right click), tool bar buttons, and 'Edit' menu items. Formerly, only Ctrl-C and Ctrl-V worked correctly. The other operations copied and pasted only the entity's name.
- **Bug fix for SubModel clone**. The inputs for the Region associated with the clone of a SubModel are now inherited correctly from the prototype SubModel. These inputs include 'Size', 'Position', 'Scale', 'Orientation', and 'Alignment'.
- **Bug fix for Assign object**. A bug has been fixed that prevented an Assign object in a SubModel clone from working correctly. Although the 'AttributeAssignmentList' input appeared to inherit its assignment input from the prototype SubModel, the input was completely ignored when entities were processed by the Assign object.

Changes in release 2023-05:

- **'Closed' input for Polyline**. If this input is set to TRUE, an extra line is added between the first and last node of the polyline.

- **'PolylineWidth' input for Polyline**. This input specifies the physical width of the polyline, measured in metres or any another DistanceUnit. The 'FillColour' specified for the Polyline determines the colour of a wide polyline.
- **'Match' input of blank or null**. An expression that returns an empty string or null for the Match input to a Server, EntityProcessor, or Seize object is now interpreted as meaning that any entity in the Queue is eligible for selection, regardless of its Match value. It is equivalent to the Match input being left blank.
- **Position, Size, Alignment, and Orientation outputs in expressions**. These outputs can now be used in expressions where they are interpreted as arrays of numbers.
- **Improved error message display for Input Builder**. The Input Builder now displays long and/or multi-line error messages in an easier-to-read fashion. The display also distinguishes more clearly between an input with a syntax error from one that is syntactically correct but cannot be evaluated at the present time.
- **Improved graphics for BarGauge and Graph**. These objects are now displayed correctly when the 2D button is turned off and they are viewed obliquely.
- **'Region' output**. All objects now have a 'Region' output which returns the Region in which the object is presently located.
- **Improved .DAE import**. Collada 3d files (.dae) generated from the FBX exporter (used by both 3ds Max and Maya software) are now imported correctly. This exporter has some quirks that are now accommodated by the JaamSim importer.
- **Bug fix for .OBJ import**. Wavefront 3d files (.obj) that use the option of relative indexing are now imported correctly.
- **Bug fix for Arrow**. The arrow head is now rotated correctly for an Arrow that is angled up or down from the horizontal.
- **Bug fix for Orientation**. A SimEntity that has an Orientation input is now displayed correctly when rotated by a Queue, EntityDelay, EntityConveyor, or EntityContainer.
- **Bug fix for Alignment**. A SimEntity that has an Alignment input is now displayed correctly by Queue and EntityContainer.
- **Bug fixes for mouse interaction in 3d space**. The selecting and rotating of objects with arbitrary 3d coordinates are handled more consistently.
- **Bug fixes for 'Show Labels' and 'Show SubModels' buttons**. These buttons now work more reliably.

Changes in release 2023-04:

- **SubModel improvements**. SubModel clones are now saved correctly to the input file. Formerly, various unnecessary and unwanted input values were saved for SubModel clones that caused input errors when the file was reloaded. Files saved with the previous release (2023-03) can be loaded successfully and can be re-saved with the correct entries.
- **Required inputs for SubModel**. Inputs for a SubModel that are normally 'required' become optional if the SubModel has one or more clones.
- **'OffsetTime' input for TimeSeries**. This input allows the event times specified for a TimeSeries object to be shifted by a fixed amount without having to adjust the times specified by the 'Value' input.
- **Input file improvements**. Unnecessary headings and white space are now longer saved to the input file.
- **Unit Selector improvements**. The menu of units offered for each unit type (TimeUnit, etc.) now includes a pop-up description for each choice. For example, the description popup for the time unit 'w' is "Weeks".
- **TraceEvents and VerifyEvents improvements**. These inputs for the 'Simulation' object now have more informative error messages that are handled correctly by the user interface.

- **Bug fix for SetGraphics**. Generated entities whose graphics are changed by a SetGraphic object are now reset correctly when they are re-cycled by an EntityGenerator. This bug was introduced accidentally in release 2023-01.

Changes in release 2023-03:

- **Improved auto-complete menus**. Auto-complete dropdown menus now prioritize the choices that have the same initial text as the user input. Choices that match the use input later in their text appear further down in the menu. This convention matches the behaviour of the auto-complete feature provided by other software packages and increase the chance that the correct item appears at the top of the menu where it can be accepted by pressing the Return key.
- **Description popups enabled for the up/down keys**. Description popups are updated when a new menu item is selected by pressing an up/down key. Formerly, a popup appeared only when the mouse was moved to a new menu item. Description popups are provided by the auto-complete menus for entities and outputs in the Input Builder as well as for its 'Function' dropdown menu.
- **Bug fix for SubModel**. SubModels can now be dragged and dropped successfully from the Model Builder. A bug introduced in 2023-02 caused drag and drop to fail for SubModels.
- **Bug fix for Input Editor**. The editor can now be scrolled up and down successfully while a model is running.
- **Bug fix for EntityConveyor**. When the 'RotateEntities' option is selected, the entities are now rotated in pitch as well as yaw as they follow a 3d conveyor path. Formerly, only the yaw rotation was performed.
- **Bug fix for EntityConveyor**. Entities are now positioned on the conveyor based on their rear instead of their middle. This change was necessary for the entities to be spaced correctly along an accumulating conveyor.
- **Bug fix for EntityLogger**. The 'TraceEntityStates' feature works correctly again. An error in release 2023-01 had caused states to be logged for additional entities.

Changes in release 2023-02:

- **Output Viewer display**. To avoid confusion, outputs are now shown in expression format: strings are enclosed by double quote and objects are enclosed by square brackets.
- **Accumulating EntityConveyor**. The EntityConveyor object has been extended to simulate 'accumulating' type conveyors such as a roller conveyor. This type of operation is controlled by the inputs 'Accumulating', 'Length', and 'AccumulationLength'. The model 'EntityConveyor Example – with Accumulating' has been added under Help > Examples to demonstrate the use of these inputs.
- **Spacing of objects on an EntityConveyor**. The input 'EntitySpace' allows the user to specify the minimum distance required on the conveyor before another object can be placed. Entities can be fed onto the conveyor with the correct spacing using the output 'ReadyForNextEntity' and by adding an ExpressionThreshold to the 'ReleaseThresholdList' input for the upstream object. If entities are placed on the conveyor without the correct spacing, they will remain at the beginning of the conveyor until sufficient space becomes available.
- **'MaxValidNumber' input for EntityConveyor**. This input limits the number of entities that can be moved by an EntityConveyor at one time. This input is intended to stop a model before it becomes unresponsive when an input error causes the number of entities on the conveyor to grow without bound.
- **SubModel reworked to use the prototype and clone system**. Clones of a SubModel are now defined in the same way as other types of clones, and the SubModelClone class has been eliminated. For example, the following inputs define a SubModel named Fred and clones named Fred1 and Fred2:

Define SubModel { Fred }
Define Fred { Fred1 Fred2 }

These changes are all backwards compatible. Old input files that use SubModelClone are still processed correctly and are re-written in the new format when the model is saved.

- **Inherited inputs for SubModel components**. The component objects of a SubModel clone now inherit their default values from the corresponding components of the prototype SubModel. Inherited values are shown in blue in the Input Editor and can be over-written by entering a new value. The new values are retained correctly when a model is saved and re-loaded. The only values that cannot be over-written are those that refer explicitly to the prototype SubModel and its components. For example, if the 'NextComponent' input to a component of a prototype SubModel named Fred is Fred.Server1, then the corresponding input for the clone SubModel Fred1 is automatically set to Fred1.Server1, but this input cannot be over-written. Inputs that are locked in this way are shown in grey in the Input Editor. If an input file is edited to over-write a locked value, an error message will be generated when the file is loaded.
- **'Save SubModel' menu item**. It is now possible to save a SubModel and its components by right-clicking on the SubModel object and selecting 'Save SubModel'. A SubModel can be saved in this way only if it does not include any external references. That is, all the inputs to the SubModel and its component objects must refer only to the SubModel and its components. An error message will be generated if it refers to an object that is not a component of the SubModel. SubModel files are saved with the extension '.sub'.
- **Import SubModel**. A SubModel can be imported to a new model by clicking on the 'Import' button in the menu bar and selecting 'SubModel'. To reflect its greater importance, the 'Import' function has been moved from the 'File' menu to its own entry in the menu bar.
- **Simulation time shown in the run-time Error popup**. To facilitate debugging, the run-time error popup now shows the exact simulation time at which the error occurred.
- **Improvements to NextList and PreviousList outputs**. These outputs can now be used in an expression.
- **Improvements to TimeSeriesThreshold**. The 'LookAhead' and 'Offset' inputs now accept expressions.
- **Bug fix for 'Utilisation' and 'Reliability' outputs**. These outputs are now calculated using the all the states defined by the 'WorkingStateList' input. Formerly, they were calculated using only the 'Working' state.
- **Bug fix for ScriptEntity**. It is now possible to specify an empty file for the 'Script' input. Formerly, an empty file would prevent model inputs from being saved if they were entered after the model has been started.
- **Bug fix for 'uniform' function**. The correct template is entered for the 'uniform' function when it is selected from the 'Function' menu in the Input Builder.
- **Bug fix for Seize object**. The 'AssignmentsAtStart' input is now executed by the Seize object. Formerly, this input had no effect.
- **Bug fix for 'Show SubModels' button**. The state of the 'Show SubModels' button is now respected correctly after a model has been reset to time zero. Formerly, all the SubModel components would be displayed after a reset regardless of the state of the button.
- **Bug fix for EntityContainer**. The entities in a generated EntityContainer are now deleted correctly when it reaches an EntitySink. The bug introduced in release 2023-01 as a result of the pooling of generated entities.
- **Bug fix for EntitySink**. The 'obj' output for an EntitySink now returns 'null'. A bug had been introduced in release 2023-01 that caused it to return the entity that had been received and then deleted immediately.

Changes in release 2023-01:

- **Object orientation by prototype and clone**. A 'clone' is an entity that inherits all its input values from another entity, called the 'prototype'. If an input value for the prototype is changed, then all of its clones inherit the new value automatically, without any copying. A clone's inherited input value can be over-written with a new value.

  A clone can inherit its inputs through a chain of prototypes and clones. For example, if the object Fred is a clone an object named Human, and the Human object is a clone of another object named Biped, then Fred will inherit all its input values from Biped except for the ones that were over-written by Human. More information about object orientation via prototype and clone can be found in the article https://en.wikipedia.org/wiki/Prototype-based_programming.

  The Input Editor shows an inherited values in blue and an over-written value in black. An inherited value can be over-written simply by entering a new value in the Input Editor for the clone. An over-written value can be cleared by deleting it, in which case the inherited value re-appears and is shown again in blue.
- **'Prototype' and 'CloneList' outputs**. These outputs show the prototype for the entity and a list of its clones. If the entity does not have a prototype, the Prototype output returns null.
- **'Define' statement extended to Prototypes**. Clones are defined in the input file using the same syntax as other objects. For example, if MyServer is an instance of Server and it has clones MyServer1 and MyServer2, then this prototype and its clones are defined in the configuration file by the statements:

  ```
  Define Server { MyServer }
  Define MyServer { MyServer1  MyServer2 }
  ```
- **Huge speedup for entity generation**. Assemble, Duplicate, EntityGenerator, EntityLauncher, and Pack have been modified to produce clones of the prototype entity instead of copies. Since a clone inherits its input values from its prototype, there is no longer a need to copy the inputs from the prototype to the clone. Furthermore, after a generated clone is destroyed by an EntitySink or other types of objects, it is reset and pooled for re-use by the prototype entity. These two factors lead to a dramatic reduction in the time required to generate an entity, from approximately 3,000 clock cycles to just 700 clock cycles. For comparison, the time to execute a schedule and execute a future event is about 240 clock cycles. Both of these times are many times faster than those for any other simulation package.
- **Object Selector updates**. The Object Selector is now updated continuously while a model runs in real time mode. When real-time mode is turned off, the Object Selector updates only when the model is paused.
- **Expression parameter 'simTime'**. The simTime parameter is equivalent to entering this.SimTime. For example, 'simTime + 5[h]' is a valid expression. Similarly, simTime can be used in an assignment input such as 'this.attrib = simTime'
- **'EnableTracing' input for Simulation**. If TRUE, the 'Trace' keyword for each object is activated. For an object to be traced, both the 'EnableTracing' input for Simulation and the 'Trace' input for the object must be set to TRUE. Trace outputs are written to standard out and can used by a programmer to track the internal logic for one or more selected objects as the model is executed. Tracing is an essential tool for testing and debugging the Java code for JaamSim.

Changes in release 2022-07:

- **Expression input to 'Show' keyword**. An expression can now be entered to the 'Show' input for any object. This feature allows the visibility of an object to be controlled by the user through a ToggleButton

or for it to be visible only when an expression returns TRUE, for example, when a breakdown occurs or when simulation time is within a range of values.

- **'Show' output**. This output returns the value of an entity's 'Show' input. It is helpful when an object's visibility is linked to another object's visibility.
- **EntityLabel visibility**. The EntityLabel for an object is now shown only when the object is visible.
- **'AssignmentTime' and 'ReleaseTime' outputs for ResourceUnit**. These outputs return the times at which the ResourceUnit was last seized and last released, respectively.
- **'InitialNumber' input for EntityGenerator**. This input instructs the generator to create the specified number of entities at time zero. This feature provides an easy way to start a model with a non-empty Queue.
- **'typeName' function**. The typeName function returns a string describing the type of the variable that is specified. It accepts any valid input and returns one of the following strings: "NUMBER", "STRING", "ENTITY", "COLLECTION", or "LAMBDA".
- **'Description' input for Group**. The 'Description' input for a Group now assigns this input to each member of the Group. A new input 'GroupDescription' can be used to provide a description for the Group itself.
- **Outputs 'Members' and 'InputValues' for Group**. These new outputs identify the entities that are members of the Group and the input keywords that have been set for the Group.
- **Bug fix for 'NumberGenerated' output**. When a new entity is generated, the NumberGenerated output is now incremented after the PrototypeEntity input is evaluated. Formerly it was incremented before the PrototypeEntity input was evaluated. This change allows the NumberGenerated output to be used more logically in an expression that is entered to the PrototypeEntity input.
- **Bug fix for inputs receiving a list of numerical expressions**. The exact text is now retained for any constant values that are entered.
- **Bug fix for 'ScriptedViewPosition' and 'ScriptedViewCenter' inputs for View.** These inputs now work correctly when new values are entered and the model is restarted.
- **Bug fix for multi-line input**. Inputs that involve include curly braces in a user-edited input file are now parsed correctly when they are spread over multiple lines.
- **Bug fix for 3d graphics file names**. File names that include invalid characters such as '#' are now trapped correctly and generate an appropriate error message.
- **Bug fix for 'TickLength' input**. This input can now be set successfully for models that include a TimeSeries object. Formerly, the presence of a TimeSeries would cause an input error when the model was reloaded.
- **Bug fix for 'EntitiesPerArrival' input**. This EntityGenerator input can now be set to zero or to an expression that returns zero.

Changes in release 2022-06:

- **Java 17**. The JaamSim executables for Windows can now be run with any version of Java, from Java 8 through Java 17. Formerly, they could not be used with Java 17.

Changes in release 2022-05:

- **'date' function**. The date function returns the calendar date and time of day for a specified simulation time. The date and time are returned as an array of integers in the format {YYYY, MM, DD, hh, mm, ss, milliseconds}.
- **'AssignmentsAtStart' input**. This new input for provides the ability to assign attribute values when a new entity is started by an EntityProcessor, Server, and many other types of objects.

- **'NextOpenDuration' and 'NextCloseDuration' outputs**. These outputs return the duration of the next open and closed conditions for a TimeSeriesThreshold.
- **Outputs 'CalculatedMin' and 'CalculatedMax'**. These outputs specify the minimum and maximum values that can be returned by a probability distribution (ignoring the 'MinValue' and 'MaxValue' inputs). For most distributions, the CalculatedMin and CalculatedMax outputs is zero and positive infinity, respectively. However, there are some notable exceptions that can cause users to make input errors, primarily the NormalDistribution (whose CalculatedMin is negative infinity) and the BetaDistribution (whose CalculatedMax is one).
- **Validation tests for 'MinValue' and 'MaxValue' keywords**. The MinValue and MaxValue inputs for a probability distribution must now satisfy the following conditions: the MinValue input must be <= the CalculatedMax output, and the MaxValue input must be >= CalculatedMin output. Formerly, inputs that did not satisfy these conditions would cause a runtime error that was difficult for users to understand.
- **Improved testing of numerical inputs**. Some inputs such as 'ServiceTime' are restricted to a specific range of values. For example, valid inputs to 'ServiceTime' are >= 0 seconds. Restrictions of this type are now checked when the input is a number, an expression, or an object that returns a number. Formerly, these restrictions were applied only when the input was a number.
- **'StateTimes' output improvement**. States in the 'StateTimes' output that have zero time are no longer displayed in the Output Viewer.
- **Bug fix for ExternalProgramServer**. Avoids a potential null pointer error when a model is restarted.
- **Bug fix for higher-order functions**. The arrays returned by the functions 'filter', 'map', 'range', and 'sort' can now be assigned to an attribute and manipulated using an Assign object. Formerly, these arrays were returned as 'read-only'.
- **Bug fix to the 'Function' menu for Input Builder**. The menu entry for the 'normal' function has been fixed to show the correct argument list. Also, the menu entries for all of the probability distribution functions have been revised to spell 'parameters' correctly.
- **Bug fix for EntitySystem**. Entries in the Event Viewer now show the object name as well as the method name ('setPresentState').
- **Bug fix for Input Editor**. The 'Input Error' pop-up now appears just once when an input is entered incorrectly in the Input Editor. Formerly, this pop-up had to be closed twice by the user and would appear unnecessarily when the Input Builder was opened to correct an error.
- **Bug fix for the Input Builder**. The 'Cancel' button now behaves correctly. A change in an earlier release had caused the edits to remain in place.
- **Bug fix for generated entities**. Entities created by an EntityGenerator now respect any 'ThresholdList', 'MaintenanceList', and 'BreakdownList' inputs that were entered to the object specified by the 'PrototypeEntity' input for the EntityGenerator. Formerly, these inputs were ignored by the generated entities.
- **Bug fix for Output Viewer**. All numerical values shown in the Output Viewer are now displayed with the same number of digits. This change is most notable for the 'StateTimes' output.
- **Bug fix for SubModel**. An error message is now generated if the user attempts to delete a SubModel that has SubModelClones that are derived from it.
- **Bug fix for Units**. The 'ConversionFactorToSI' input for a user-defined unit is now saved in the correct position in the input file.
- **TimeSeriesThreshold improvement**. Keyword descriptions for 'MaxOpenLimit' and 'MinOpenLimit' have bee revised to make their meanings clearer.

- **Random distribution improvements**. The descriptions for the 'GlobalSubstreamSeed' and 'RandomSeed' keywords now use the terminology 'random stream' instead of 'seed'. The use of the word 'seed' was misleading in this context. The descriptions have also been revised for the probability distribution functions in the 'Function' menu for the Input Builder.
- **'NextEntity' keyword improvement**. The description of this keyword has been revised to add a note that the processor does not monitor the value of the input expression. It must be notified using the 'WatchList' input.

Changes in release 2022-04:

- **Probability distribution functions**. Functions have been added to the expression systems that return random samples from the various probability distributions. For example, the function 'normal' returns a sample from a normal distribution. The functions can be entered with or without an explicit value for its random seed. If the seed is omitted, JaamSim chooses the first unused seed value for the function. The arguments for each distribution are given in its pop-up description provided by the Input Builder under 'Function'.
- **Discrete Distributions**. New probability distribution objects have been added for the most commonly used discrete probability distributions: binomial, discrete uniform, geometric, negative binomial, and Poisson.
- **'ExternalProgramServer' object**. This object is similar to the ExternalProgram object, but achieves much higher performance by leaving the external program open and passing messages, instead of opening and closing the external program each time it is used. The new object requires an overhead of only 0.1 milliseconds to execute an external Python program compared to 250 milliseconds for the ExternalProgram object. ExternalProgramServer is fast enough that it can be executed one million times by a model without a significant increase in run time.
- **'Next Time' button for Event Viewer**. The 'Next Time' button has been modified to leave the model at the present simulation time after executing all the future events and conditional events for that time. Formerly, the simulation time would be advanced to the next event time.
- **'WatchList' and 'VerifyWatchList' keywords for ExpressionLogger**. An ExpressionLogger can now operate more efficiently by specifying a WatchList input. Note that this feature is only used with the ValueTraceList input.
- **'WatchListCondition' input for ExpressionLogger**. This input allows an ExpressionLogger to log selected changes to the objects in the WatchList. If the expression entered for this input returns TRUE then an entry is made to the log using the normal format. If it returns FALSE, then the change to the WatchList object is ignored.
- **'WatchedEntity' output for ExpressionLogger**. This output returns the object in the WatchList that triggered the most recent entry in the log. When used in an expression entered to the WatchListCondition, it returns the object that triggered the potential log entry. If the expression returns FALSE then the WatchedEntity value is reset to its previous value.
- **'LogEntity' output for ExpressionLogger and EntityLogger**. This output returns the entity that triggered the most recent log entry. It is useful when the TraceEntityStates feature is used with the EntityLogger object.
- **TRUE and FALSE in expressions**. Expressions can now contain the strings TRUE and FALSE to indicate the values 1 and 0 respectively.
- **Expression inputs for run parameters**. The 'RunDuration' and 'InitializationDuration' inputs can now accept expressions.

- **'PresentTime' output for Simulation**. This output returns the elapsed time since the epoch (1997-01-01 00:00:00). This output is useful for determining the execution time for a model.
- **Bug fix for Pack**. An error introduced in version 2022-03 has been corrected. This error caused the NumberToStart input to be set to 1, regardless of the number that was entered.
- **Bug fix for 'Prototype' input**. An error in an expression entered to the Prototype input for EntityGenerator is now handled correctly. Formerly, an error would cause the software to lock up.

Changes in release 2022-03:

- **Improved runtime error messages**. An expression input that causes an error during a simulation run generates an error message popup that includes the offending expression, the location of the error, and the name of the keyword that was affected.
- **Improved validation error messages**. An expression input that fails the validation checks on start-up now generates an error message with the same information as a runtime error.
- **Window size and position adjustment for Windows 11**. The sizes and positions of the various JaamSim windows have been adjusted for a better fit under Windows 11. These adjustments are the same as those for Windows 10.
- **Improvements to the Text object**. The DataSource input can now handle an expression that returns a number that has dimensions, without setting the Units input. For example, input of '1[kg] + 2[kg]' generates a display of 3[kg]. Formerly, this worked only if Format was left blank. It now works correctly with any valid input to Format. For example, a Format input of 'mass = %s' would generate a display of "mass = 3[kg]".
- **DataSource input can handle null**. The DataSource keyword for a Text object can now handle an input that returns null. Formerly, a value of null would generate the 'Input Error' display.
- **More accurate mean and standard deviation**. The algorithm for calculating the mean and standard deviation values has been changed to 'Welford's Online Algorithm' which reduces accumulated round-off error.
- **Bug fix for Pack and AddTo objects**. The 'NumberOfEntities' and 'NumberToStart' keywords now work correctly when a random distribution is entered to either input. Formerly, a distribution would be sampled every time an entity arrived, which distorted the statistics. These inputs are now sampled just once for each EntityContainer.
- **Bug fix for WaitQueue keyword**. This keyword generates a useful error message if an expression is entered that causes an error. Formerly, this would cause the program to lock up.
- **Bug fix for multiple selection of objects**. An object cannot be selected if it is linked by 'RelativeEntity' inputs to another object that has already been selected. For example, if an EntityLabel is selected then it is not possible to select simultaneously the object it labels.
- **Bug fix for 'indexOfMinCol' and 'indexOfMaxCol' functions**. These function now return a useful error message if the array input includes any non-numeric values. Formerly, the input would be deleted in the Input Editor without any feedback.
- **Bug fix for Unpack and RemoveFrom**. These objects now reset correctly to time zero. Formerly, it was possible for these objects to end a simulation run in a state that was not completed cleared when reset to time zero. This bug could occasionally result in different (but valid) simulation results for a run that was restarted after being reset.

Changes in release 2022-02:

- **'ProcessPosition' keyword for SubModel**. When the components of a SubModel are not displayed, the entities being processed by the components are now displayed at the location specified by ProcessPosition input for the SubModel. Formerly, the entities would be displayed at the locations of the hidden sub-model components.
- **Bug fix for 3d graphics**. A bug has been fixed that prevented the View windows from being displayed for some Windows computers.
- **Bug fix for HiDPI monitors**. The text shown in the Input Builder and Log Viewer is now scaled correctly for high DPI monitors (e.g. 4K monitors).
- **Bug fix for CustomOutputList**. Input to the CustomOutputList keyword is now updated to reflect any name changes to the objects in the expression.
- **Bug fix to AutoSize for Text objects**. The auto-size feature now works correctly when a text object is edited in place.
- **Bug fix for error messages**. Pop up dialog boxes for error messages no longer cause deadlocks when a model is started.
- **Bug fix for log files**. A log file (.log) is now saved whenever an error occurs in a simulation run. The log file contains the same information as displayed in the Log Viewer.
- **Bug fix for headless mode**. The log file is now saved correctly in headless mode.
- **Bug fix for SubModelStart and SubModelEnd**. These objects now work correctly with the 'Create Entity Links' and 'Show Entity Flow' buttons.

Changes in release 2022-01:

- **Error Message for Failed Runs**. An error message popup is now displayed at the end of a series of simulation runs if one or more had a run time error.
- **Bug fix for Resources**. A queue can now be shared successfully by multiple resources users such as the Seize and EntityProcessor objects. Formerly, a run time error could occur in this situation.
- **Bug fix for Run Progress**. The Run Progress display is now updated only when the model is running. Formerly, the display could update itself while the model was paused, causing it to show a much lower execution rate.
- **Bug fix for NumberOfThreads**. A configuration file that sets an input to its default value is now handled correctly when replications on multiple threads are being prepared.

Changes in release 2021-06:

- **'NumberOfThreads' keyword for Simulation**. This input specifies the number of simulation runs to perform simultaneously while executing a series of scenarios and replications. Typically, this input can be set to the number of CPU cores. For example, an input of 4 will give good performance for a CPU with four cores such as the Intel i7. A larger input can also be specified with only a small decrease in performance due to additional thread management by the computer.
- **Run Progress display**. This display is shown when multiple scenarios and/or replications are started and 'Real Time' mode is off. It shows the scenario and replication numbers for each of the threads being executed along with the overall percent complete for the set of simulation runs to be executed.
- **Overall Progress**. The percent completed for the entire set of scenarios and replications is now shown in the title bar. Formerly, it showed the percent complete for the present simulation run.

- **'RunParameterList' keyword for Simulation**. This input specifies a list of quantities to be printed to the custom output report, similar to the RunOutputList keyword. The difference is that these quantities specify the inputs to each scenario and therefore the 95% confidence intervals are not reported.
- **Log Viewer cleanup**. Some of the unnecessary error messages shown in the Log Viewer have been suppressed. These messages were related to the graphics for the Queue and EntityContainer objects.
- **Additional outputs for ResourceUnit**. ResourceUnit now includes all the outputs provided by the Resource and ResourcePool objects.
- **'SeparateFiles' keyword for EntityLogger and ExpressionLogger**. This input specifies whether the logs for multiple runs should be saved to separate files or appended to a single file.
- **Runtime Errors recorded for multiple runs**. When multiple runs are to be executed, the full set of runs is completed even if runtime errors occur in one or more of the runs. The error messages are reported in both the .dat output file and the Log Viewer. Formerly, a runtime error would generate a pop-up error message that stopped any further progress.
- **'TraceEvents' and 'VerifyEvents' inputs**. The TraceEvents input for Simulation instructs the model to generate an output file containing the exact sequence and timing of all the discrete events that were executed. The VerifyEvents input instructs the model to compare its events to those in the event file that was generated previously by the TraceEvents input. These two inputs can help a programmer determine what causes two simulation models to generate different results when they were expected to be identical.
- **'-optional_graphics' or '-og' tag**. This tag suppresses the pop-up error message that is normally generated when a computer is unable to generate the 3d graphics needed for the View windows.
- **Bug fix for File inputs**. The Input Editor now shows the input value a file input as a normal file path that can include spaces, e.g. `'Test File.dat'`. Formerly, it was shown as an encoded file path, where spaces were replaced by the code %20, e.g. `Test20%File.dat`.
- **Bug fix for expressions**. An expression that returns a null entity is now handled correctly.
- **Bug fix for ResourceUnit maintenance and breakdowns**. Forced maintenance and forced breakdowns now work correctly when the same DowntimeEntity is used in the ForcedMaintenanceList or ForcedBreakdownList input for more than one ResourceUnit.
- **Bug fix for 'UniqueMatchValues'**. This output for Queue and Container now returns the correct list of values.
- **Bug fix for 'RandomSeed' input**. The RandomSeed input for a probability distribution contained in a sub-model clone is now saved to the input file. This input ensures that same random seed is used when the model is re-loaded.

Changes in release 2021-05:

- **User Manual added to the Help menu**. The User Manual can now be accessed through the software by selecting Help > User Manual in the menu bar. An internet connection is not required to view the manual.
- **Simplified names for copied objects**. When an object to be copied has its default name, its copy is assigned the next available default name. For example, if there are two Servers named 'Server1' and 'Server2', then a copy of Server1 is named 'Server3'. Formerly, it would be named 'Server1_Copy1' and its copy would be named 'Server1_Copy1_Copy1' and so on.
- **'ToggleButton' added to Basic Objects**. This object provides a button that can be clicked by the user to turn it on or off.
- **Fill and outline colours for Text objects**. Text objects can now be provided with background fill and/or outline colours using the same keywords as Shape objects.

- **'AutoSize' keyword for Text objects**. When this keyword is set to TRUE, the Text object's size is adjust to match the size of the text that has been entered. Auto-sizing is performed whenever the text is edited or its font or text height is modified.
- **Revised 'Return' key action for the Input Editor**. The Input Editor no longer wraps to the first input when the 'Return' key is pressed on the last input.
- **Model Builder popups re-positioned**. Popups for objects in the Model Builder now appear on its right hand side. Formerly, they appeared under the mouse pointer which partially obscured the next object below.
- **Object Selector popups**. Popups have been added to the Object Selector that show the input to the 'Description' keyword for each object.
- **Dialog box for graphics errors**. An error message dialog box now appears if the 3D graphics system does not start up normally. Formerly, graphics system errors were reported only to the Log Viewer.
- **Bug fix for Polyline objects**. A filled polyline can now be dragged when the mouse pointer is anywhere over the filled area. Formerly, it could only be dragged when the mouse was over its outline.
- **Bug fix for Input Editor popups**. A bug has been fixed that occasionally prevented a keyword popup from appearing.
- **Bug fix for the 'Align Right' and 'Align Left' buttons**. These buttons now work correctly for Text objects that have been rotated from the horizontal. Formerly, the Text object would jump to a new position when the 'Align Left' or 'Align Right' buttons were pressed.
- **Bug fix for TimeSeriesThreshold warnings**. Warning messages that a TimeSeriesThreshold will always be closed have been revised to provide the correct information.
- **Bug fix for Feb. 29**. February 29 is now recognized as a valid date that can be used in the data file for a FileToVector, FileToMatrix, or FileToHashMap object. Note that if February 29 is used in a non-leap year or in any year with the GeorgianCalendar input set to FALSE, it will be interpreted as March 1 for that year.

Changes in release 2021-04:

- **'NumberOfReplications' keyword**. This input allows the user to specify the number of replications to perform for each scenario.
- **'StartingScenarioNumber', 'EndingScenarioNumber', and 'ScenarioIndexDefinitionList' inputs**. The inputs that specified the number of simulation runs have been converted to specify the number of scenarios to simulate. Old input files that specified the number of simulation runs will be converted automatically to specify an equivalent number of scenarios, each with a single replication to perform.
- **'ScenarioNumber', 'ScenarioIndex', and 'ReplicationNumber' outputs**. These new outputs are analogous to the 'RunNumber' and 'RunIndex' outputs. The latter outputs have been retained to maintain backwards compatibility.
- **Mean Values and Confidence Intervals for each Scenario**. The mean values for the outputs specified by the RunOutputList are now printed automatically for each scenario when multiple replications are performed. The 95% confidence intervals for the outputs are also reported.
- **'PrintReplications' and 'PrintConfidenceIntervals' keywords**. These inputs allow the user to control the contents of the output report. In many cases, it will be more convenient to print just the mean values and confidence intervals for each scenario and to suppress the outputs for each replication.
- **'PrintRunLabels' keyword**. When set to TRUE, this input inserts a column in the run output report that shows the scenario number for each simulation run. If the 'PrintReplications' input is also TRUE then a second column showing the replication number is also added.

- **Simplified inputs for multiple runs**. The default settings for the 'ScenarioIndexDefinitionList' and 'GlobalSubstreamSeed' keywords have been modified so that these inputs are not required in most cases.
- **Expression inputs for multiple runs**. The StartingScenarioNumber, EndingScenarioNumber, and NumberOfReplications inputs now accept expressions as well as numerical values.
- **'TraceEntityStates' keyword**. This input allows the EntityLogger object to trace the subsequent state changes for entities that pass through this object. An example model has been added to demonstrate the use of this new feature.
- **'NumberOfDuplicates' keyword**. This input allows the user to specify the number of duplicate entities to send to each of the targets for a Duplicate object.
- **Entity Finder improvements**. A number of improvements have been made to the Entity Finder tool to make it easier to use.
- **Integer-valued inputs in Input Editor**. Inputs that are restricted to dimensionless integers, such as Capacity for Resource, are now shown without decimal digits in the Input Editor. Formerly, an input value of '1' would be displayed as '1.0'.
- **Output Viewer actions Ctrl-F and Ctrl-C**. Keyboard shortcuts for 'Find' and 'Copy' have been added to the Output Viewer.
- **Bug fix for 'QueueLengthTimes'**. The array returned by this Queue output now includes the full histogram of queue lengths from zero length to the maximum observed length. Formerly, it included only the entries for the minimum length to the maximum length.
- **Bug fix for Queue**. The positions of the entities in a queue are now calculated using their present sizes. Formerly, they were calculated using the input values for their sizes.

Changes in release 2021-03:

- **'Show References' button**. This button generates arrows that indicate the objects referenced in the inputs to the selected object(s). If no object is selected, the arrows are shown for all the displayed objects in the model.
- **'EntityReferenceList' output**. Returns a list containing the objects that are referenced in the inputs to this object.
- **'Presentation Mode' button**. Closes the tool windows and expands the active view window to fill the computer screen. When the button is de-activated, the tool and view windows return to their original sizes and positions.
- **'ConsumedEntityList' output for Combine and Assemble**. Returns a list of the entities that were destroyed during the last combine or assemble action.
- **'Concurrent' keyword for DowntimeEntity**. Enables the downtime activity to start while one or more other maintenance or breakdown activities are already in progress. Normally, only one breakdown or maintenance activity can be performed at any given time.
- **'Create Entity Links' button with calculation objects**. The inputs to a calculation object can now be set using the Create Entity Links button in the same way as the process flow objects. Clicking on one calculation object followed by another sets the second objects inputs for 'Controller', 'UnitType', and 'InputValue'.
- **Bug fix for 'Create Entity Links' button**. A link arrow is shown from the selected object only when it able to form a link. For example, an arrow is no longer shown when an EntitySink is selected.
- **Bug fix for probability distributions in a SubModelClone**. A bug has been fixed that caused the RandomSeed input to a probability distribution within a SubModelClone to change value each time a model is restarted.

- **Bug fix for log files**. The log files generated by the EntityLogger and ExpressionLogger objects are now closed correctly when a user closes JaamSim prior to the completion of a simulation run.
- **Bug fix for '3D Graphics Example'**. The 3d object in this example model is now shown correctly.

Changes in release 2021-02:

- **Drag and drop for SubModel**. The 'Pre-built SubModels' entry in the Model Builder now includes entries for any SubModels created by the user. When one of these SubModels is dragged and dropped, a SubModelClone object is generated that behaves in the same way as a copy of the SubModel, but has the important additional property that it can be updated to remain consistent with its parent SubModel. The updating occurs automatically when a model is started or manually by selecting 'Update Clones' from the context menu for the parent SubModel.
- **Multi-line Text and OverlayText**. Text and OverlayText objects with multiple lines of text can now be edited in place using the usual mouse and keyboard actions. A new line can be inserted in the text by pressing Ctrl-Enter. Editing can be initiated by double clicking on the object or by selecting it and pressing F2. Formerly, only single line Text and OverlayText objects could be edited in place.
- **'SelectionCondition' and 'NextEntity' inputs**. These new inputs for Server, Seize, EntityGate, and EntityProcessor allow these objects to choose a particular object from its queue. The SelectionCondition input accepts an expression that returns a boolean that determines whether an entity is eligible to be remove from the queue. The NextEntity input accepts an expression that returns a specific entity to remove from the queue. These inputs can be used individually or combined with each other and with the Match input to specify the entity to process. A new example model 'Fill Order Example – with NextEntity and SelectionCondition' demonstrates the use of these inputs.
- **'ReleaseThresholdList' input for EntityProcessor**. The ReleaseThresholdList input works for EntityProcessor in the same way that it works for Server. When a threshold in the list closes, each of the EntityProcessor's units continues to work until the entity being processed is ready to be released, at which point the unit is halted. The other units continue to work until they too are ready to release their entity. When the threshold re-opens, all the waiting entities are released at the same time. A new example model 'EntityProcessor Example – with ReleaseThresholdList' demonstrates this behaviour. Similar examples for Server and EntityConveyor were also added.
- **'KeywordList' input for SubModel**. This input provides a way to define new inputs for a SubModel. Each of these inputs accepts an expression that returns a number with or without units. An output with the same name as the input is also created to return the value for the input expression. This feature allows a component of the SubModel to obtain all its inputs through its parent SubModel or from other components of the SubModel, thereby avoiding any references to objects outside of the SubModel. The inputs and outputs created by the KeywordList input for a SubModel are retained by any dragged and dropped SubModelClones of that SubModel.
- **Animated 3D Graphics**. The 'Actions' keyword for ColladaModel allows the 3D motions created in the imported 3D file (.DAE, .JSB, etc.) to be reproduced by any DisplayEntity that uses this ColladaModel. The motions imported with the 3D file are listed by the 'Actions' output for the ColladaModel. A new example '3D Animation Example' demonstrates a model with multiple independent 3D motions.
- **'UserList' output for Thresholds and DowntimeEntity**. The new output lists the objects that include the Threshold or DowntimeEntity in one or more of their ThresholdList, DowntimeList, or MaintenanceList inputs.
- **Attributes and custom outputs in the Output Viewer**. The attributes and custom outputs that have been defined for an object are now shown at the bottom of the Output Viewer under the heading for the

object type. Formerly, attributes and custom outputs were shown near the top of the Output Viewer, under the heading for 'Entity'.

- **Labels for SubModel components**. Labels are now shown for the components of a SubModel when the components are visible and the 'Show Labels' button is selected in the user interface.
- **Entity flow arrows for SubModel components**. Entity flow arrows are now shown for SubModel components when the components are visible and the 'Show Entity Flow' button is selected in the user interface.
- **Example models from the website**. The four example models displayed on the JaamSim website (3D Graphics Example, Server and Queue Example, Cafe Model Example, and Harmonic Oscillator Example) have been included in the Examples dialog box.
- **Bug fix to entity flow arrow for 'Duplicate'**. Entity flow arrow are now shown for the all the entity flows from a Duplicate object.
- **Bug fix for string inputs**. A string input is now tested for the presence of a single quote. Formerly, a single quote would appear to be accepted correctly but when the model was saved, its input file would be unreadable.
- **Bug fix for array- and map-valued attributes**. This bug fix ensures that the values stored by an array- or map-valued attribute are independent of any copies of the object. Formerly, the Duplicate object created copies whose array- and map-valued attributes were linked to one another.
- **Bug fix for ColladaModel**. A bug has been fixed that had prevented some imported 3D models from being selected using the mouse.
- **Bug fix for JSB format**. A bug has been fixed that prevented the use of the JaamSim binary format (.jsb) for 3D models.
- **Bug fix for File inputs**. The name of an input file can now be edited directly in the Input Editor without using the file browser. Formerly, a confusing error message would appear when a relative file path was modified.

Changes in release 2021-01:

- **Examples dialog box**. A set of example models are now provided within JaamSim. An example model can be loaded by clicking on Help > Examples and selected the model from a list. The Examples dialog box includes a field for searching the example models by name as well as a preview window that shows an image of the model.
- **Label position**. For a polyline type object (EntityConveyor, etc.), the default label position has been changed to a point below the mid-point between the first and last nodes. Formerly, the label was positioned below the first node. This change allows the default position to be used for most polyline type objects. However, it shifts the positions of any polyline object label saved with earlier versions of JaamSim. The best way to correct an earlier model is to delete the labels for the polyline object and re-create them again using the 'Show Labels' button.
- **Error dialog for multi-line expressions**. The expression error dialog box now shows an expression on multiple lines when it was entered in this format. The location of the error condition is indicated by the start of red coloured text.
- **Improvements to the Change Graphics dialog box**. The preview window in the dialog box now uses a more suitable field of view (FOV) and scale to show 3d objects.
- **Bug fix for imported image and 3d files**. Image and 3d files that contain a period are now handled correctly. For example, a zipped Collada file named 'filename.dae.zip' will now generate an entity named

'filename'. Formerly, the entity would be named 'filename.dae' which caused an error message to be generated because the dot is supposed to indicate a component of a sub-model.

- **Bug fix for time stamp error messages**. The error message for an incorrect time stamp value is now shown correctly. Formerly, the error message would trigger another error and generate a misleading message.
- **Bug fix for Control Panel menus**. Arrow keys can now be used with the various drop-down menus provided by the Control Panel.
- **Bug fix for scrollable menus**. Scrollable menus now work correctly with the up and down arrow keys. Formerly, a menu did not scroll automatically to show the selected entry when the arrow keys were pressed.
- **Bug fix for Input Editor**. Fixes a bug introduced in release 2020-13 that prevented the 'Edit' option from working correctly in the input error dialog box.

Changes in release 2020-13:

- **Input Editor data entry.** The Input Editor now uses the same conventions as Excel for entering and editing input values. The arrow keys, F1, F2, Ctrl+C, Ctrl+V, Ctrl+Z, and Ctrl+Y have all been implemented (i.e. help, edit, copy, paste, undo and redo).
- **Input Editor retains last keyword**. When a new entity is selected, the Input Editor automatically selects the keyword that had been selected for the previous entity. Formerly, the Input Editor tab would be retained but not the keyword.
- **Newlines retained when entered in the Input Builder**. An expression entered over multiple lines in the Input Builder is now saved and re-opened in that format. Formerly, an input entered on multiple lines would be collapsed to a single line when it was accepted from the Input Builder.
- **Simplified boolean inputs**. In addition to accepting TRUE and FALSE, boolean inputs can now accept t, T, or 1 for TRUE and f, F, or 0 for FALSE when entered through the Input Editor. A single character input is converted to the standard TRUE or FALSE values after it is accepted.
- **Auto-correction for expression inputs**. Inputs that accept an expression now enclose the input with single quotes automatically when required. The single quote marks are added after the input is accepted in the Input Editor or Input Builder.
- **Spacebar can be used to select from the auto-complete menu**. When an auto-complete menu appears in the Input Builder or other tools, the highlighted entry can be accepted by pressing the Spacebar key or the Return key. Formerly, only the Return key could be used.
- **Look and feel for Linux**. JaamSim now uses Java's standard cross-platform look and feel when run under Linux. This is the same look and feel used under the Max OS X operating system. Windows is the only operating system under which JaamSim uses the native look and feel.
- **FirstTime and Interval keywords for Controller**. These keywords now accept expressions. Formerly, they were restricted to numerical values.
- **NumberPending output for DowntimeEntity**. This output provides information on the number of backlogged maintenance activities. A DowntimeEntity can have backlogged activities if it is entered in the OpportunisticMaintenanceList or OpportunisticBreakdownList inputs for Server and other types of objects.
- **Present Time and Date**. The 'Present Time and Date' output for the Simulation object has been renamed to 'PresentTimeAndDate' so that it can be used in an expression. A number of other outputs have also been renamed to remove the spaces in their former names.

- **SimDate and SimDayOfWeek outputs for Simulation**. These outputs provide the numerical values for the time, date, and day of week at the present simulation time.
- **PresentDate and PresentDayOfWeek outputs for Simulation**. These outputs provide the numerical values for the time, date, and day of week at the present real time.
- **Improved error messages for CustomOutputList**. The expression entered when defining a custom output is now checked when possible that it returns the correct unit type. Name conflicts with attribute definitions and the built-in outputs are also detected.
- **Bug fix for Input Builder**. Cursor position is unchanged after the down arrow key is pressed to select from the auto-complete menu. Formerly, the cursor location would move to the next line after the menu item was selected.
- **Bug fix for log file**. Log entries are now ignored if the log file cannot be opened. This can occur when another model has opened a log file with the same name or an existing log file is open in another program such as NotePad++.
- **Bug fix for preferred units**. Preferred units are now stored separately for each model that has been opened. Formerly, models with different preferred units could interfere with each other.

Changes in release 2020-12:

- **Multiple Models**. Two or more models can now be opened at the same time. A 'Window' entry has been added the menu bar to allow switching between models. A model can be started and switched to the background while it continues to execute. At present, it is not possible to copy an object from one model and paste it to another.
- **Help window**. A new Help window shows documentation from the user manual when a topic is selected. The Help window can be opened through the Help menu or by pressing F1.
- **Multi-line Text objects**. The Text and OverlayText objects can now display text on multiple lines. One or more newline characters can be added to a string by entering %n whenever a new line is to appear. For example, the string "Quick red%nfox" will be displayed with "Quick red" on the first line and "fox" on the second line.
- **Improvement to link arrows**. A link arrow is now shown between the centres of two objects when there is insufficient space to show an arrow between them. Formerly, a link arrow was not shown when there was insufficient space.
- **Bug fix for launching JaamSim**. JaamSim now launches reliably to its full size. Formerly, it would occasionally launch to its minimized state.
- **Bug fix for Turkish language**. A bug has been fixed that caused a null pointer error and a severe slowdown when a Graph object was dragged and dropped into a model that was being executed on a computer that was set to the Turkish language.
- **Bug fix for copying a SubModel**. The inputs to the components of a SubModel are now converted correctly when the SubModel is copied. Formerly, the conversion could be done incorrectly depending on the SubModel's name.
- **Bug fix for NextMaintenanceTime and NextBreakdownTime outputs**. These outputs are now returned with the units of Time. Formerly, they were returned as dimensionless values.
- **Bug fix for renaming**. Renaming an entity by editing its label now behaves correctly when the new name conflicts with an existing name. Formerly, the entity was left in an inconsistent state after the error was detected.

Changes in release 2020-11:

- **Improvements to the Input Builder**. The auto-complete menus for object names and output names now show a selected item, which can be accepted by pressing the Enter key or scrolled using the up/down arrow keys. Formerly, it was necessary to use the mouse to select any of the menu items.
- **ImageModelDefault** object. A default ImageModel object is now provided for use with the OverlayImage object. Formerly, an OverlayImage was invisible until its DisplayModel input was set.
- **JaamSim icon**. Higher resolution versions of the JaamSim icon are now available when needed on computers that display large sized icons.
- **Find window**. The window for the Find tool now has a separate entry in the Task Bar and retains its last position when closed.
- **Bug fix for moving multiple entities**. A single undo entry is now generated when multiple entities are moved. Formerly, the undo list would contain a large number of incremental moves for each object.
- **Bug fix for entity flow arrows**. Entities flow arrows are now shown between Assemble and Combine objects and their Queues.
- **Bug fix for Model Builder**. The state of the Model Builder's tree display is retained when an object is renamed. Formerly, the tree was collapsed when an entity was renamed.
- **Bug fix for Output Viewer's pop-ups**. The output description pop-ups are now shown only when the cursor is located in the output name column, and not when it is in the output value column.
- **Bug fix for 'GeorgianCalendar' and 'StartDate' keywords**. These keywords can now be used when a data file for a FileToVector/Matrix/HashMap object includes entries in YYYY-DD-MM format.
- **Bug fix for 'StartDate'**. The StartDate keyword now works correctly for dates prior to 1970-01-01.

Changes in release 2020-10:

- **Multiple entity selection**. Multiple entities can now be selected by pressing the Control key and clicking on the objects to be selected. Once selected, the entities can be moved as a group using either the mouse of the arrow keys.
- **'Show Entity Flow' button state**. The state of the 'Show Entity Flow' button is now retained when a model is saved and reloaded.
- **'BaseName' keyword for Pack and Duplicate**. The name for the ContainerEntities generated by Pack and the copied entities generated by Duplicate can now be specified by the user.
- **Bug fix for Pack, Duplicate, Assemble, and EntityLauncher**. The default names for the entities generated by these objects are now valid when they are located in a SubModel.
- **'MatchRequired' and 'NumberRequired' keywords for Combine**. These keywords work in the same way as they do for the Assemble object. The Assemble and Combine objects now work in exactly the same way. The only difference is that the Assemble object generates a new entity while the Combine object re-uses the entity from the first queue. **IMPORTANT NOTE: The MatchRequired keyword must be set to TRUE for the Combine object to work in the same way as previous releases.**
- **'FirstQueue' keyword for Combine and Assemble**. This keyword determines which match value to use when several values have the required number of entities.
- **Bug fix for Input Builder**. Clicking the 'Cancel' button in the Input Builder no longer adds an entry to the redo list.
- **'Add Node' menu item**. A new node can now be added to a polyline type object using the context menu (right click). Formerly, the only way to add a node was to use the keyboard shortcut Ctrl + Alt + Left Click.

Changes in release 2020-09:

- **Bug fix for 'Create Entity Links' button**. A link is created for a dragged and dropped entity only when the 'Create Entity Links' button is depressed. Formerly, the link was always created regardless of the state of the 'Create Entity Links' button.
- **Bug fix for Seize**. A large number of resource units can now be seized at one time. Formerly, a run time error would occur if more than 10 units were seized at once.

Changes in release 2020-08:

- **Arrow in 'Create Entity Links' mode**. When the 'Create Entity Links' button is selected an arrow is shown between the selected entity and the present mouse position. This feature reminds the user that the model is in 'Create Entity Links' mode.
- **Link created on drag and drop**. When an entity is dragged and dropped in 'Create Entity Links' mode, a link will be created from the selected entity to the newly created entity. This feature allows the user to connect the model automatically as objects are dragged and dropped.
- **'hh:mm' format**. Times can now be entered in 'hh:mm' and 'YYYY-MM-DD hh:mm' formats. Formerly, it was necessary to include the input for seconds, i.e. 'hh:mm:ss' and 'YYYY-MM-DD hh:mm:ss', even when the seconds value was zero.
- **Colour menus**. Named colours are now grouped by colour family, e.g. 'Red colours', 'Blue colours', etc.
- **Additional named colours**. The full set of X11 colours is now available.
- **EntitySystem**. The new WatchList keyword for EntitySystem is now used to specify its components. The EntitySystemList keyword has been deleted.
- **'ContainerStateAssignment' keyword for Unpack**. This keyword allows a state to be assigned to the EntityContainer being unpacked.
- **'Container' output for Unpack**. This output returns the EntityContainer that is being unpacked or null if no container is present.
- **Bug fix for VideoRecorder**. A view showing a rotated display is now captured correctly.
- **Bug fix for Assemble and Combine**. When sufficient entities are available for more than one match value, the match value is chosen for the entity that has been waiting the longest.
- **Bug fix for Pack and Unpack**. The outputs NumberAdded, NumberProcessed, and NumberInProgress are now calculated correctly for Pack and Unpack. For both objects, it is the entities being packed/unpacked that are counted, not the EntityContainers.

Changes in release 2020-07:

- **'MaxValidLength' keyword for Queue**. This new keyword places a maximum value on the number of objects in the Queue. An error message is generated if the maximum length is exceeded. This input is intended to prevent JaamSim from becoming unresponsive when an input error causes a queue to grow without bound – a common error for beginners. It has no effect on model logic. The default value of 10,000 is large enough for most simulation models.
- **Unique 'RandomSeed' input**. The random seed input for a copied probability distribution is reset to a unique value. To avoid unwanted correlations between probability distributions, each RandomSeed input in a simulation model should have a unique value.
- **Probability Distribution drag and drop**. The delay between the first probability distribution to be dragged and dropped and the appearance of the new object has been eliminated.
- **Bug fix for 'ReleaseThresholdList' keyword**. A bug has been fixed that caused a runtime error when the ReleaseThresholdList input for Server was used.

- **Bug fix for 'CustomOutputList' keyword**. An object with a CustomOutputList definition that references another CustomOutputList value for the same object can now be copied successfully.
- **Bug fix for ExpressionThreshold**. A bug has been fixed that caused a runtime error when an ExpressionThreshold object referenced the state of another ExpressionThreshold.

Changes in release 2020-06:

- **'SubModel' object**. A new SubModel object is now available to be dragged and dropped into a model. A Region for the SubModel components is created automatically along with the SubModel object. Objects that are dragged and dropped onto the SubModel's region become components of the SubModel.
- **'SubModelStart' and 'SubModelEnd' objects**. These objects are used to specify the entry and exit points for a SubModel. An entity arriving to a SubModel is directed automatically to its SubModelStart component (a SubModel can contain only one SubModelStart component). An entity arriving to a SubModelEnd component is directed automatically back to the parent SubModel and from there to the object specified by its NextComponent input. An entity can be sent directly to any other object by providing an input to the NextComponent input to the SubModelEnd object.
- **Nested SubModels**. A SubModel can contain other SubModels, which in turn can contain other SubModels.
- **Copying a SubModel**. A SubModel object can be copied in the same way as other objects. The SubModel's components and their inputs are copied along with the SubModel. Inputs to SubModel components that reference the parent SubModel or other SubModel components are changed automatically to refer to the copied SubModel or SubModel component. RandomSeed inputs to probability distributions are changed to the first unused integer value.
- **Copying an object to a SubModel**. An object that is copied to a position within a SubModel's region becomes a component of that SubModel in the same way as a dragged and dropped object.
- **Inputs to ServerAndQueue components are saved**. Inputs set by the user to the components of a ServerAndQueue object are now saved to the .cfg file.
- **Unnecessary keywords deleted for ServerAndQueue**. 'RegionScale', 'RegionSize', 'RegionPosition', and 'ServiceTime' have been deleted because these inputs can be applied directly to the Region and Server components and their values are now saved in the input file.
- **Locked inputs to components of ServerAndQueue**. Inputs to the components of a ServerAndQueue object that are set automatically, such as the WaitQueue input to the Server, are now locked in the Input Editor to prevent them from being over-written by the user. The Input Editor displays a locked input in grey text.
- **Bug fix for File objects**. The DataFile input to a file object (FileToVector, FileToMatrix, and FileToHashMap) is now ignored if it is left blank. This change fixes a bug that prevented the API method setValue from being used successfully to set the data for a file object.

Changes in release 2020-05:

- **Execution Time Profiler**. The Event Viewer now includes an extra tab that shows the percent of execution time that is consumed by each type of event. Note that a model must be run at full speed with 'Real Time' off to get accurate event durations.
- **Mac OS-X Look and Feel**. The look and feel on Mac computers has been changed to solve a number of serious deficiencies in the user interface under this operating system: the ability to see the state of a button, the size of the buttons in the Input Builder, and the lack of lines in the Input Editor. The revised look and feel is less Mac-like but much more usable than the default system.

- **'length' function**. A new function has been added to the expression system that returns the number of characters in a specified string.
- **Sub-model components in dropdown menus**. The components of a sub-model such as ServerAndQueue are now included in the dropdown menus for model inputs.
- **Bug fix for Input Builder**. A bug has been fixed that caused the last model input to be undone if the Input Builder is open and closed without making any changes.
- **Bug fix for Assign**. An assignment to an array is now performed correctly when the index is a calculated value. Formerly, an error message was generated if the new entry was not dimensionless.
- **Bug fix for MatchValue output**. The MatchValue output for Server, Seize, etc. is now displayed correctly by a Text object when the output is null (the default value).
- **Bug fix for Seize and EntityProcessor (MatchValue output)**. The MatchValue output is now set correctly by the Seize and EntityProcessor objects.
- **Bug fix for Seize and EntityProcessor (WatchList input)**. The WatchList input now works correctly for the Seize and EntityProcessor objects.
- **Bug fix for Event Viewer**. The Event Viewer can now be opened and closed safely while a model is running at full speed with 'Real Time' turned off. Formerly, this would cause the user interface to lock until the run had been completed.
- **Bug fix for AttributeDefinitionList**. The input to this keyword is now tested to ensure that there are no duplicated attribute names.

Changes in release 2020-04:

- **Bug fix for re-sizing.** The bug introduced in release 2020-03 that prevented the re-sizing of an object has been corrected.

Changes in release 2020-03:

- **Bug fix for 3D Graphics**. A bug has been fixed that caused computers with AMD graphics processors to be unable to display the icons for some objects. This bug was introduced accidentally with the ColladaModel optimisation in release 2020-01.
- **'Safe Graphics' tag**. The command line tag '-sg' instructs JaamSim to use only the simplest features in OpenGL. This tag is intended for users with older computers who have been unable to get a view window to open correctly.
- **Expression input to 'WaitQueue'**. Servers and related objects can now accept an expression input to their WaitQueue keywords. If the expression can return several different queues during the simulation run, it is necessary to enter all of the possible queues in the input to the WatchList keyword.
- **'WatchList' keyword for Server, etc.** This input accepts a list of objects that will be monitored by the Server or related object. Whenever one or more of the monitored objects changes state, the Server will wake-up and test its queue for a suitable entity to process. This feature can be used to allow the Server to process entities from multiple queues and to be alerted to a change to its Match value input.
- **'WatchList' keyword for ExpressionThreshold**. This input accepts a list of objects that will be monitored by the ExpressionThreshold. If this input is provided, the ExpressionThreshold evaluates its OpenCondition and CloseCondition expression inputs and set its open/closed state ONLY when triggered by an object in its WatchList. This is much more efficient than the default behaviour which evaluates these expressions at every event time and whenever its state is queried by another object.
- **'VerifyWatchList' keyword for ExpressionThreshold**. Allows the user to verify that the input to the 'WatchList' keyword includes all the objects that affect the ExpressionThreshold's state. When set to

TRUE, the ExpressionThreshold uses both the normal logic and the WatchList logic to set its state. An error message is generated if the threshold changes state without being triggered by a WatchList object.

- **'ScaleFactor' keyword for NonStatExponentialDist**. This input allows the arrival rate for a non-stationary exponential distribution to be scaled up or down. Note that it is incorrect to scale the inter-arrival times returned by the distribution, hence the need for this new keyword.
- **Input Editor and Output Viewer popups**. The popups describing the input keywords for the Input Editor and the output descriptions for the Output Viewer are now shown on the right hand side of these tool windows. This change prevents the popup from obscuring the keywords/outputs names and prevents the popup from becoming stuck which could occur sometimes.
- **'Previous' and 'Next' buttons**. These new buttons allow the user to select the next or previous object in the chain of linked objects that follow the flow of entities in the model. This feature can be very useful when inspecting a complex model where the entity flow is not obvious.
- **Bug fix for headless mode**. Headless mode now works correctly. It had been broken accidentally in release 2019-08.

Changes in release 2020-02:

- **Input Builder improvements**. The auto-complete feature for the Input Builder is now available for more complex expressions and has been extended to support SubModel features.
- **SubModel components included in main report**. Outputs for SubModel components are now included in the main report.
- **'Controller' keyword for ExpressionThreshold**. This new keyword allows an ExpressionThreshold to be updated at only the times specified by a separate Controller object. Normally, it is updated whenever it is tested and after each event time, which can become costly in a large simulation model with frequent events.
- **Controller object**. The Controller object has been moved to the Basic Objects palette and its functionality extended with new keywords 'FirstTime' and 'MaxUpdates' and new outputs 'EntityList' and 'Count'.
- **EntitySystem object**. This new object allows the state for a group of objects to be assigned by the user. A new input 'StateEntity' has been added to most objects to allow them to be included in the group of objects for a particular StateEntity. By default, the EntitySystem's state is set to 'Working' if any of the objects in the system are working, and to 'Idle' if all the objects are idle. An optional keyword 'StateExpression' allows user-defined states to be assigned to the EntitySystem depending on the states of the individual object in the system.
- **Bug fix for graphics**. An error has been fixed that had caused JaamSim to crash for some older computers that do not support OpenGL 4.3. This bug was introduced in release 2020-01.
- **Bug fix for selecting objects**. An error has been fixed that had caused some shapes of objects to be difficult to select with the mouse. This bug was introduced in release 2020-01.
- **Bug fix for an apostrophe in text**. Text objects cannot include an apostrophe (single quote) because this symbol is used as in delimiter in JaamSim's input system. Users are now prevented from entering an apostrophe when a text object is edited.
- **Bug fix for times in 'hh:mm:ss.sss' format**. Times entered in the 'hh:mm:ss.sss' format are now parsed correctly. This bug was introduced in release 2019-10.
- **Bug fix for API method 'save(File)'**. The specified file is now saved to the correct directory.

Changes in release 2020-01:

- **SubModel components**. SubModel components now use dots instead of percent signs in their naming convention, e.g. 'SubModel1.component1' instead of '%SubModel1%component1'. The new system supports SubModels within SubModels.
- **'Parent' output for Entity.** The parent of a SubModel component is the SubModel object.
- **Keyword 'sub' for expression.** The keyword 'sub' in an expression refers to the parent SubModel. It is equivalent to entering 'this.parent'.
- **Region set for dragged and dropped entity.** When an entity is dragged and dropped on top of a region, the entity's Region input is set to that region.
- **Copying a SubModel**. The components of a SubModel are copied and pasted automatically when the SubModel is copied.
- **Bug fix for ExpressionLogger**. The headings for the log file always appear above the log entries. Formerly, it was possible for state changes at time zero to be recorded BEFORE the headings.
- **Error messages for ExternalProgram**. The entire error message now appears in the error pop-up if an error occurs in an external program. Formerly, only the first line of the error message appeared.
- **Polyline object**. A new Polyline object has been added to the Graphics Objects palette. A Polyline can be shown with or without a fill colour and with or without an outline colour.
- **'Filled', 'FillColour', and 'Outlined' keywords for PolylineModel**. Polylines can now be filled with a specified colour and they can appear with or without their outline.
- **'DisplayModel' and 'Edit DisplayModel' buttons**. Buttons have been added to the Control Panel to allow an object's DisplayModel to be changed or its present DisplayModel inputs to be edited.
- **'Clear Formatting' button**. This Control Panel button resets all the inputs on the Format tab for the selected entity to be reset to their default values.
- **Arrow keys for polyline type objects**. Polyline type objects can now be moved using the arrow keys in the same way as normal objects.
- **Entity links for EntityLauncher and Assemble**. The PrototypeEntity input for EntityLauncher and Assemble can now be set using the 'Create Entity Links' button in the same way as EntityGenerator.
- **Bug fix for Assemble, AddTo, and Combine**. Formerly, these objects caused graphical problems if the 'Show Entity Links' button was selected before their queue inputs were assigned.
- **'BaseName' keyword for Assemble**. This new keyword for Assemble works in the same way as the BaseName keywords for EntityGenerator and EntityLauncher.
- **Enhancements to Entity Flow arrows**. Entity flow arrows are now shown more consistently between entities that are close to one another. They are also shown in blue to help distinguish them from model components.
- **Keystrokes sent to EntityLauncher**. EntityLauncher(s) in a model now detect the keystroke specified by their ActionKey input without being selected.
- **Pasting using the context menu**. After an entity is copied, it can be pasted by right-clicking anywhere in the model and selecting 'Paste'. Formerly, the right-click (context) menu did not appear unless an entity was selected.
- **Bug fix for pasting**. The position of a pasted object now respects its RelativeEntity and Region inputs. Formerly, a copied object with either of these inputs would not appear at the location at which it was pasted.
- **NumberOfUnits keyword** for Seize and Release. The input to the NumberOfUnits keyword can now be entered as a series of numbers, e.g. 1 2, instead of { 1 } { 2 }. Furthermore, the default value of 1 is now applied to all the resources defined in the ResourceList input.

- **NumberOfUnits keyword for Assemble.** This input now accepts expressions using the same format as the NumberOfUnits keyword for Seize and Release.
- **String functions.** The expression system now includes the following new functions for manipulating strings: 'substring', 'indexOfStr', 'toUpperCase', 'toLowerCase', 'trim', and 'split'. These functions mirror the string functions provided in the Java programming language.
- **Bug fix for Input Builder.** The Input Builder now accepts any expression that is accepted by the Input Editor. Formerly, an expression such as 'this.obj.attrib' would not be accepted by the Input Builder if the 'obj' output was null, but would be accepted by the Input Editor.
- **ResourcePool outputs.** ResourcePool now has the same outputs as the Resource object.
- **ColladaModel optimization.** Complex Collada models are now displayed much more efficiently on computers with an Nvidia or AMD graphics processor. Frame rates for these models are now as much as 10 – 20 times faster with a powerful graphics processor.

Changes in release 2019-11:

- **Entity Finder**. This new tool allows an entity to be selected by entering its name. The tool can be displayed using a new button on the button bar or by pressing Ctrl+F.
- **Copy and Paste**. Entities can now be copied and pasted.
- **SimTime column for Logger**. Simulation time is now displayed without units to make it easier to use in a spreadsheet. The column heading has been changed to 'this.SimTime/1[unit]', where [unit] is the preferred unit for time.
- **'Edit' menu**. A new Edit menu has been added with entries for Undo, Redo, Copy, Paste, and Delete.
- **'MaxEntitiesToDisplay' keyword**. This new keyword for Simulation allows the user to specify the maximum number of entities to display in the view windows. This limit prevents JaamSim from becoming unresponsive when the number of entities in a model exceeds the graphics capabilities of the computer. The default limit is 10,000 entities which is sufficient for all but the most complex models.
- **Object Selector limit removed**. The Object Selector can now display an unlimited number of entities. Formerly, it was limited to 10,000 generated entities.
- **Keyboard shortcuts for Run, Pause, and Speed**. The space bar can now be used to run or pause the simulation model. In 'Real Time' mode, the '>' and '<' keys can be used to increase and decrease the simulation speed respectively. Note that it is not necessary to press the Shift key when using these two shortcuts.
- **Keyboard shortcuts for New, Open, Save, Save As, and Exit**. The standard keyboard shortcuts have been implemented for these functions.
- **Bug fix for Reset**. Fixes a bug introduced in release 2019-09 that prevented the 'Reset' button from working correctly for models that included an EntityContainer.
- **Bug fix for Input Builder**. The present simulation time is now used to calculate the Present Value display in the Input Builder. Formerly, the displayed value assumed a simulation time of zero.
- **Bug fix for undo**. Input changes that generate an error message are no longer included in the undo list. Formerly, it was possible for this to occur under some circumstances.
- **Bug fix for context menu**. The context menu generated by right-clicking on an entity in the Object Selector is now displayed in its correct position.

Changes in release 2019-10:

- **'GregorianCalendar' keyword**. This new input to Simulation allows the user to choose between the standard Gregorian calendar that includes leap years and a simplified calendar that has a fixed 365 days per year. Formerly, models were restricted to the simplified calendar.
- **'StartDate' keyword**. This new input to Simulation allows the user to specify the calendar date and time of day that corresponds to zero simulation time. This input is used throughout the model to convert date inputs to simulation time. Formerly, only the year displayed by OverlayClock could be set by the user and this input had no effect on the interpretation of other date inputs to the model.
- **'UnitTypeList' keyword removed**. The UnitTypeList keyword has been removed from Simulation, ExpressionLogger, and EntityLogger. It is best to have these objects record dimensionless outputs, but if an output with dimensions is recorded the value will appear with the appropriate unit.
- **'ValuePrecisionList' keyword removed**. The ValuePrecisionList input has been removed from ExpressionLogger. The number of decimal places in an output can be set using the 'format' function.
- **'sum' function**. This new function returns the sum of the entries in an array or map that contains numbers.
- **'Description' keyword**. This new keyword allows additional information about the entity to be entered.
- **Options tab**. A number of keywords have been moved to the 'Options' tab in the Input Editor. These include the 'AttributeDefinitionList', 'CustomOutputList', 'Active', and 'StateAssignment' keywords that are used by many types of entities. Also moved to the Options tab are the 'PauseCondition', 'ExitAtPause', and 'TickLength' keywords for the Simulation object.
- **Multi-line expressions**. A complex expression can now be entered on two or more lines in the input file.
- **Comments in expressions**. Expressions can now include one or more comments to make long expressions more readable. Any text entered between two hash marks ('#') is interpreted as a comment. For example, '1 + 2 #first comment# + 3 #second comment#' returns 6.
- **'OffsetToFirst' keyword**. This new keyword for TimeSeries determines how a time stamp entry in the time series is converted to simulation time. When set to TRUE, the simulation time corresponding to a time stamp is calculated as the elapsed time between the time stamp and the first time stamp in the series. This option ensures that the first entry in the TimeSeries corresponds to zero simulation time. If set to FALSE, the simulation time for the time stamp is the elapsed time between the time stamp and the 'StartDate' input for the Simulation. The default value for OffsetToFirst is TRUE so that existing models are not affected.
- **Negative times in TimeSeries**. If the time stamp for a TimeSeries entry is before the 'StartDate' input, the corresponding simulation time is negative. The logic for TimeSeries and its 'CycleTime' input have been adapted to allow for this possibility.
- **Assignment to map attributes**. The Assign object can now assign an entry in map-valued attribute. The syntax for assigning to a map is similar to that for assigning to an array: '[ent].attrib("key") = someExp', where ent is the entity, attrib is the entity's map-valued attribute, key is the string-valued key for the map, and someExp is an expression that return the value to be assigned. Note that the "key" can be replaced by an expression that returns a string.
- **'UserList' output for Resources**. This output lists the objects that are able to seize this resource.
- **'ResourceList' description**. The description for this keyword now includes information on how to assign the priority for seizing a resource
- **Bug fix for 'Real Time' mode**. This bug introduced in release 2019-09 caused models to execute more slowly than specified in Real Time mode.

- **Bug fix for 'AttributeDefinitionList' and 'CustomOutputList' keywords**. This bug introduced in release 2019-05 caused an error message to be generated when the input to either keyword was deleted.
- **Bug fix for EntityProcessor**. This bug affected an EntityProcessor whose ResourceList input was left blank. In this case, the EntityProcessor did not restart correctly when a threshold re-opened or when downtime was completed.

Changes in release 2019-09:

- **'ExternalProgram' object**. This object executes an external program when it receives an entity. Model execution is suspended until the external program terminates. The external program can be written in Python or any other language.
- **'parseNumber' function**. This new function converts a string to a dimensionless number.
- **Map definition**. It is now possible to define an new map within an expression using the syntax '{ "string1"=value1, "string2"=value2, etc. }'. The values can be a number, string, entity, etc. or any valid expression. This feature allows an attribute to be defined whose value is a map.
- **Optimisation for huge models**. JaamSim has been optimized for models with 100,000 or more entities. Entity creation and destruction is approximately four times faster for models with this number of entities. Models with as many as 300,000 entities have been tested with acceptable processing speeds. There is no apparent limit to the number of entities that can be modelled efficiently given sufficient RAM.
- **Improvements to file inputs**. The file browser is now initialized to the presently selected file or, if no file is selected, to the folder containing the configuration file.
- **Bug fix for simultaneous runs with thresholds**. Models that include thresholds (such as ExpressionThreshold) now work correctly when executed simultaneously using the API. Formerly, the thresholds could allow interactions between simultaneous runs that caused a time error to be generated.
- **Bug fixes for Event Viewer**. The Event Viewer is now closed when 'Close All Tools' is selected. Furthermore, the Event Viewer now retains its present state when a model is minimized and then restored.
- **Bug fix for inputs accepting a list of expressions**. An error message is generated if an input such as RunOutputList receives more than one expression inside a single pair of curly braces. Formerly, the extra inputs were deleted automatically without any message.

Changes in release 2019-08:

- **'Show Labels' button changes**. The 'Show Labels' button now distinguishes between 'permanent' and 'temporary' labels, and shows/hides only the temporary labels. A permanent label is one that was created individually through an entity's right-click menu.
- **'Show SubModels' button changes**. The 'Show Sub-Models' button now distinguishes between 'permanent' and 'temporary' sub-model displays, and shows/hides only the temporary sub-model displays. A permanent sub-model display is one that was activated individually through the sub-model's right-click menu.
- **Keyboard shortcuts for undo, redo, and save**. The standard keyboard shortcuts Ctrl+Z, Ctrl+Y, and Ctrl+S can now be used to perform the undo, redo, and save-model actions. Note that these shortcuts are only available when working in a view window. They do not work from the Input Editor or other tool windows.
- **Automatic region assignment for SimEntity**. When a SimEntity is passed to an object such a Server, it is now assigned automatically to that object's region and is scaled appropriately for that region.
- **FileToHashMap**. This new object is differs from FileToVector and FileToMatrix in that its Value output returns a hash map of the data in the file. Each record in the file generates a separate 'key' and 'value' in

the hash map. The first entry in the record is the key, which must be a string. The remaining entries are placed in an array that becomes the value for the key.

- **'setValue' methods for File objects**. The API now includes methods for passing data to the FileToVector, FileToMatrix, and FileToHashMap objects without using a data file.
- **Bug fix for Object Selector**. A bug has been fixed that occasionally caused the Object Selector to show a blank entry for the Simulation object after a model has been loaded.
- **Bug fix for dragging within a scaled Region**. A bug has been fixed that affected the dragging behaviour for some objects when contained in a Region whose Scale input is less than one.
- **Bug fix for Queue and EntityContainer in a scaled region**. The entities contained by a Queue or EntityContainer are now positioned correctly when in a region whose Scale input is less than one.
- **Bug fixes for 'Lock 2D' button**. An oblique view is always shown when the button is de-selected. Formerly, this was not always the case. Also, when more than one view window is present, the button is applied to the view window that is active. Formerly, it was applied to the last window that was created.
- **Bug fix for 'Centre in View' menu item**. Formerly, this feature did not always work correctly.
- **Bug fix for 'Reset Positions and Sizes'**. A bug introduced in release 2019-07 has been fixed that prevented the 'Reset Positions and Sizes' menu item from working after a model has been loaded.

Changes in release 2019-07:

- **Larger view widow**. The default sizes and positions of the various JaamSim windows have been adjusted to maximize the size of the view window. The JaamSim windows now fill the entire screen.
- **Multiple monitors**. The JaamSim tool windows (Input Editor, etc.) now follow the control panel when it is dragged to second monitor.
- **'Show Labels' button**. A new button has been added to the control panel that shows/hides the labels for every object in the model. A new object that is dragged and dropped to the model will acquire a label automatically when the button is selected.
- **'ReleaseThresholdList' keyword**. When a threshold in the ReleaseThresholdList input closes, the object continues to work normally until it is ready to release an entity. At this point, the object stops work without releasing the entity. The entity is released as soon as the threshold re-opens (assuming that nothing else such as a breakdown or another threshold prevents the object from working). The ReleaseThresholdList input has been added to Server and EntityConveyor in the present release. Additional objects will acquire this input in subsequent releases.
- **'ServerAndQueue' object**. A new 'SubModel Objects' palette containing a new 'ServerAndQueue' object has been added to the Model Builder. The 'ServerAndQueue' object is intended to provide an example of how a sub-model can be programmed. It consists of a Server, its Queue, and an ExpressionThreshold that closes when a maximum queue length is reached.
- **'Show SubModels' button**. A new button has been added to the control panel that shows/hides the components of every sub-model.
- **Valid entity names**. The set of prohibited characters for an entity name has been increased to include every character that causes problems. The full set consists of spaces, tabs, curly braces, single or double quotes, square brackets, and hash characters.
- **Cannot delete a generated entity**. Users are now prevented from deleting an entity that has been generated by an EntityGenerator or any other object.
- **'Region' and 'RelativeEntity' keywords for Region**. A region can now exist within another region and be positioned relative to another entity.

- **Improved keyword pop-ups for Seize, Release, and Resource**. The pop-ups now indicate that the inputs for NumberOfUnits and Capacity must be integers.
- **Error, Integral, and Derivative outputs for PIDController**. These new outputs show the internal calculations performed by the PIDController.
- **Bug fixes for PIDController**. The OutputUnitType keyword now works correctly and the DerivativeValue output is displayed correctly.
- **Bug fix for tool windows**. The sizes and positions of the windows such as the Model Builder are now displayed more reliably.
- **Bug fix for entity flow display**. The entity flow arrows are now displayed correctly for an object whose Region input has been set.
- **Bug fix for validation errors**. It is no longer possible to start a simulation with a validation error. This bug was introduced accidentally in release 2019-03.
- **Bug fix for the secondary axis for Graph**. The SecondaryUnitType input now works correctly.

Changes in release 2019-06:

- **Region keyword**. When a Region is deleted, all the entities in that region now have their Region input reset.
- **Pack, Unpack, AddTo, and RemoveFrom**. Formerly, an OperatingThreshold closure, ForceMaintenance event, or ForcedBreakdown event would cause these objects to stop packing/unpacking entities as soon as the present entity was finished. Now, they stop when the EntityContainer is finished packing/unpacking and has been released.
- **Round function**. The 'round' function has been added to the expression system.
- **Bug fix for displayed simulation time**. Simulation time is now displayed correctly as zero after the reset button is clicked. This bug was introduced accidentally in release 2019-04.
- **Bug fix for EntityProcessor**. An entity being handled by an EntityProcessor is now moved into position from the queue as soon as its resources are seized.
- **Bug fix for AllowOvertaking**. This keyword is used by EntityDelay to ensure that entities are released in the same order as they were received. Formerly, round-off error could cause entities that are scheduled for release at the same time to be released in the wrong order.
- **Bug fix for Pack and AddTo**. If a Pack or AddTo object is packing an EntityContainer but has stopped because its queue is empty, the closure of an ImmediateReleaseThreshold will now cause the EntityContainer to be released.
- **Bug fix for Input Builder**. Windows for other applications can now appear over the Input Builder.
- **Bug fix for Input Builder**. The Input Builder is now set automatically to the correct width. Formerly, on some computers, the Input Builder was not wide enough to show the Accept and Cancel buttons.

Changes in release 2019-05:

- **Scale keyword for Region**. This new keyword applies a constant factor to the local coordinates within the Region. For example, an input value of 0.5 causes everything in the region to appear half as large.
- **Entity name cannot be blank**. Setting an entity name to an empty string generates an error message.
- **AttributeDefinitionList and CustomOutputList keywords**. If required, braces are now added automatically around the inputs to these keywords.
- **Bug fix for mismatched braces**. Inputs consisting of a series of entries enclosed by braces are now checked to ensure that there are equal numbers of opening and closing braces. Formerly, it was possible

to enter an input with mismatched braces in the Input Editor that would be interpreted correctly at the time but would corrupt the input file.

- **Bug fix for DisplayedUnits**. The input to the DisplayedUnits keyword is now read correctly in headless mode (-h). Formerly, this input was ignored in headless mode.
- **Bug fix for ResourceUnit**. The state for a Resource unit is now set correctly to "Stopped" when it is halted by a Threshold.
- **Bug fix for EntityContainer**. The entities in an EntityContainer are now positioned correctly when it is rotated.
- **Bug fix for Pack, Unpack, AddTo, and RemoveFrom**. The ImmediateReleaseThreshold input now releases the container that is being processed.

Changes in release 2019-04:

- **API**. Multiple instances of JaamSimModel can now be executed simultaneously on separate threads.
- **API methods**. The following methods have been added to the API:

`getSimTime()`	Returns the present simulation time in seconds.
`isRunning()`	Returns true if the simulation is executing.
`defineEntity(String, String)`	Creates a new entity with the specified type and name.
`setInput(String, String, String)`	Sets the input for the specified entity and keyword to the specified string.
`getStringValue(String)`	Evaluates the specified expression and returns its value as string.
`getDoubleValue(String)`	Evaluates the specified expression and returns its value as a double.
`save(File)`	Writes the inputs for the simulation model to the specified file.

- **Multi-index assignment**. An element of a multi-index array can now be set using the Assign object. For example, the AttributeAssignmentList keyword can now accept the following type of input 'this.attrib(2)(3) = 5', where the attribute 'attrib' is a two index array. Formerly, assignment was restricted to a single index array. Note that each index in the assignment input can be replaced by an expression that returns a number. Furthermore, the object on the left hand side prior to the dot (".") can be replaced by an expression that returns a object.
- **indexOf**. The indexOf method now returns zero if the specified value is not found in the array. Formerly, an error message was generated.
- **WaitForEntities keyword for Pack and AddTo**. This input determines whether to move an EntityContainer into position as soon as one is available or to wait until sufficient entities are available to start loading.
- **NumberToRelease keyword for EntityGate**. This input determines the maximum number of entities to release each time the gate opens.
- **MaxRows keyword for Queue and EntityContainer**. This input allows a Queue or EntityContainer to display its entities as a 3D stack. The length and width of each layer in the stack is determined by the MaxPerLine and MaxRows inputs.
- **NextStartTime output for DowntimeEntity**. This output provides the time at which the next downtime event will begin.

- **NextMaintenanceTime and NextBreakdownTime**. These outputs provide the times for the next maintenance activity and the next breakdown, respectively.
- **NextOpenTime and NextCloseTime for TimeSeriesThreshold**. These outputs provide the times at which the threshold will open and close, respectively.
- **Bug fix for Input Editor**. A bug has been fixed in the Input Editor that prevented some inputs from being accepted automatically when the user clicks on another object or tab. This bug was most noticeable for Boolean inputs.

Changes in release 2019-03:

- **API**. JaamSim has been restructured to introduce an Application Programming Interface based on the class 'JaamSimModel'. At present, only one JaamSimModel can be created and executed at a time. This restriction will be lifted in a future release.

Constructors:

`JaamSimModel()`	Creates an new instance of a JaamSim simulation model.

Methods:

`autoLoad()`	Creates the built-in objects such as Simulation, Units, etc.
`configure(File)`	Loads the specified input file. The autoLoad() method must be executed first.
`start()`	Starts one or more simulation runs as specified by the input file.
`pause()`	Suspends model execution at the present simulation time.
`resume(double)`	Resumes a paused model. The model will continue execution until the specified simulation time at which the model will be paused. Events scheduled at the pause time will not be executed until the model is resumed.
`reset()`	Sets the simulation time to zero and re-initialises the model. The start() method can be used to begin a new simulation run.
`clear()`	Deletes all the objects in the present model and prepares the JaamSimModel to load a new input file using the autoLoad() and configure() methods.

Changes in release 2019-02:

- **ShowEntities keyword for Queue**. This new keyword allows the entities in a Queue to be hidden. It works in the same way as the ShowEntities keyword for EntityContainer.
- **RotateEntities keyword for EntityDelay**. This new keyword allows the entities in the EntityDelay to rotate to the direction of the polyline. It works in the same way as the RotateEntities keyword for EntityConveyor.
- **Active keyword for Loggers**. The EntityLogger and ExpressionLogger objects can now use the Active keyword. This provides an easy way to turn off logging when it is not needed.
- **Bug fix for UI freezing**. This bug fix prevents the user interface from becoming unresponsive when a model creates and destroys huge numbers of entities very rapidly.
- **Bug fix for entity rotation**. An entity that is rotated as it flows through a simulation model is now rotated with respect to its initial rotation as specified by its Orientation input.
- **Bug fix for ExpressionThreshold**. This fixes a bug introduced in 2018-07 that caused models to work incorrectly when a ExpressionThreshold was closed and re-opened at the same simulation time.

Changes in release 2019-01:

- **Bug fix for expressions**. The '==' operator now works correctly with objects. Formerly, objects could be compared correctly in isolation, but could not be combined with other Boolean operations such as && and ||.
- **'null' object**. The keyword 'null' can now be used in expressions. With this change, the usual test for null, 'notNull(ent)' can be written more clearly as 'ent != null'.
- **Negative Spacing**. The Spacing keyword for Queue and EntityContainer now accepts a negative value. This option allows large numbers of entities to be shown in less space.
- **EntityList output**. The new EntityList output for EntityConveyor and EntityDelay shows the entities that are being processed.
- **RotateEntities keyword**. This new keyword for EntityConveyor causes the entities handled by the conveyor to be rotated so that they point in their direction of travel.
- **Bug fix for EntityProcessor**. A bug has been fixed that allowed an EntityProcessor to seize resources when it was stopped by a breakdown and maintenance activity.
- **Bug fix for EntityContainer**. Entities that were hidden by setting ShowEntities to FALSE for the EntityContainer are now made visible again when they are removed.
- **Bug fix for Queue**. The entities placed in a queue are now rotated relative to their initial orientation. Formerly, the rotation assumed that their initial orientation was zero.

Changes in release 2018-09:

- **Shape**. A new 'Shape' object has been added to the Graphics Objects palette. This object makes it easier to add simple graphical elements to a model such as rectangles and circles.
- **Line and Shape formatting**. Buttons have been added to the Control Panel for selecting the colour and width of lines, and the outline and fill properties of Shape objects.
- **Overlay mouse interaction**. An Overlay object can now be selected and re-positioned using the mouse. When an Overlay object is dragged-and-dropped, it is positioned at the selected location in the window.
- **OverlayText editing**. An OverlayText object can now be edited in place in the same way as a normal Text object.
- **Validation**. Model inputs are now validated automatically when an input file is first loaded. Formerly, these tests were performed only when a model was started, and an error message was returned for only the first test that failed. The validation process checks that all the required inputs have been provided and tests whether the various inputs are consistent with one another.
- **Bug fix for InputValue**. Fixes a bug introduced in release 2018-08 that prevented an InputValue object from being edited.
- **Bug fix for ExpressionLogger**. Prevents a duplicate entry in the trace file at time zero when the ValueTraceList keyword is used.

Changes in release 2018-08:

- **Text formatting**. Buttons and input fields have been added to the Control Panel for setting the font, colour, and height of a selected text object. These controls work for overlay and billboard text as well as for normal text objects. The effect of a font or colour change is previewed while scrolling through the drop down menus with the mouse.
- **Position of overlay text**. Overlay text can be re-positioned using the arrow keys. First, select the overlay text object using the Object Selector and then click on the title bar of the view window.

- **Keyboard inputs to a dragged and dropped object**. A dragged and dropped object can now accept keyboard inputs (such as the arrow keys and the delete key) as soon as it is placed in a view window. Formerly, it was necessary to click on the object before it would accept any keyboard inputs.
- **Releasing zero resource units**. A Release object can now accept an input of zero to its NumberOfUnits input.
- **Error message when a file cannot be opened**. An error dialog box is now shown when an input file or log file cannot be opened. This situation can occur when a network blocks access to the file.
- **Bug fix to Priority and Match inputs for EntityContainer**. An expression containing 'this.obj' now works correctly when it is entered to the Priority and Match inputs to an EntityContainer. This bug was introduced accidentally in release 2018-07.

Changes in release 2018-07:

- **ResourcePool and ResourceUnit**. These new objects represent a major extension of JaamSim. In contrast to the existing 'Resource' object, which models a collection of identical resource units, a ResourcePool is a collection of unique ResourceUnits that have individual characteristics. An entity that enters a Seize object can select the most appropriate ResourceUnit to seize using each unit's AssignmentCondition and AssignmentPriority inputs. The AssignmentCondition input determines whether the entity is eligible to seize the unit. The AssignmentPriority input determines which of the eligible units is most appropriate for the entity. Inputs for thresholds, maintenance, and breakdown determine whether the unit is available to be seized. A ResourceUnit will attach itself to the entity that seized it if the FollowAssignment input (on the Graphics tab) is set to TRUE. The appearance of a ResourceUnit changes to indicate its present state. By default, it changes colour, but the StateGraphics keyword (on the Graphics tab) allows it to change to another shape or image.
- **Active keyword**. The ResourceUnit object includes a new 'Active' keyword that determines whether the unit can be used for the present simulation run. Other JaamSim objects will acquire this input in future releases.
- **Snap to grid for resizing and rotating**. Objects that are resized or rotated using the mouse now respect the Snap to Grid setting. The increment for rotation is fixed at one degree.
- **Control Panel buttons**. New buttons and input fields have been added to the Control Panel for formatting Text objects and for setting the snap grid spacing. The text formatting buttons allow font, style (bold or italic), and alignment to be set. Additional buttons for text height and colour will be added in the next release.
- **Polygon and star shapes**. A number of new polygon and star shapes have been added to ShapeModel. The new polygon shapes are Pentagon, Hexagon, and Octagon. The new star shapes are Pentagram, Heptagram, and Octagram.
- **Warnings for deletion**. An error message will appear if the user attempts to delete an object that is non-movable or that was defined prior to the RecordEdits command in the input file.
- **Bug fix for Queue**. The ReleaseTime output is now set correctly for Queue.
- **Outputs for TimeSeries**. New outputs NextTime and NextValue have been added TimeSeries.
- **Improvements to ExpressionLogger**. The ValueTraceList keyword now accepts all types of expression, not just the ones that return a number.

Changes in release 2018-06:

- **Pause Time**. The Pause Time input is now ignored if it is less than or equal to the simulation time. This change allows a run to be paused at the specified time and then resumed by pressing the Play button. Formerly, the Pause Time would have to be cleared to resume the run.
- **Play button disabled at end of run**. The Play button is now disabled at the end of a simulation run. It is no longer possible to run a simulation beyond its end time.
- **Speed display**. The 'Achieved Speed Multiplier' display now shows the correct speed when multiple runs are performed.
- **StateAssignment expression**. An expression can now be entered as the input to the StateAssignment keyword. Formerly, only a constant string could be entered.
- **SetEntityState keyword for EntityContainer**. If the SetEntityState is TRUE, the states for the entities contained by the EntityContainer are set to the same state as the EntityContainer every time its state is changed. If FALSE, the entities retain their original states.
- **ContainerStateAssignment keyword for Pack and AddTo**. An EntityContainer is assigned this state when it arrives to the Pack or AddTo object. Note that the StateAssignment input determines the state of the arriving entities, not the EntityContainer.
- **InitialState keyword for SimEntity**. Determines the state of the SimEntity at the start of the simulation run or when it is first created during a simulation run.
- **Default for Format keyword**. The default value for the Format keyword for Text objects is now '%s'. Now that the 'format' function can be used in expressions, the string shown by a Text object can be prepared entirely in DataSource input without the use of the Format keyword. It is recommended that the Format, UnitType, and Unit inputs for Text are left blank and all formatting is done by the expression entered to DataSource.
- **Input interpreted as a string**. An input to a keyword that expects a string or an expression returning a string will be interpreted as a string if it cannot be parsed as an expression that returns a string. For example, the DataSource keyword will now accept 'The quick brown fox' with or without the single quote marks even though the input contains spaces.
- **BaseName keyword for EntityLauncher**. The keyword BaseName has been added to EntityLauncher. It works the same as the BaseName keyword for EntityGenerator.
- **GraphicalLength output**. A new 'GraphicalLength' output has been added to all JaamSim objects. For a polyline type object such as EntityDelay and EntityConveyor, this output returns the length of the polyline as determined by its Points and CurveType inputs.
- **Bug fix for headless mode**. A bug that prevented headless from working correct for some models has been corrected. The bug affected models in which any of the model building tools such as the Input Editor or Input Builder had been resized.

Changes in release 2018-05:

- **AllowOvertaking and MinSeparation keywords for EntityDelay**. These keywords provide optional restrictions on the movement of entities with respect to one another. Setting the AllowOvertaking keyword to FALSE prevents an entity from leaving the EntityDelay before the previous entity. Setting the MinSeparation keyword to a non-zero time duration (with AllowOvertaking set to FALSE) prevents an entity from leaving the EntityDelay within the specified duration after the previous entity.
- **Match, Priority and FIFO keywords for EntityContainer**. The Priority and FIFO keywords allow an EntityContainer to store its entities in a particular sequence, while the Match keyword allows an identifier to be attached to the entities that are loaded. These keywords work for EntityContainer in exactly the

same way that they work for Queue. The entities removed from the EntityContainer can be restricted by providing an input to the new MatchForEntities keyword for the Unpack and RemoveFrom objects.

- **ResetEntityStateTimes keyword for Statistics**. The state times for entities that cycle through a closed path of objects can be collected by using the Statistics object by setting its ResetEntityStateTimes keyword to TRUE. This input causes the entity's StateTimes data to be reset to zero when it leaves the Statistics object.
- **Drop-down menu for named colours**. For colour inputs, the Input Editor now provides a list of named colours from which to select.
- **Input Editor automatically loads inputs**. When using the Input Editor, changing the tab or selecting a new object now causes an edited keyword input to be loaded without pressing the Enter key. Formerly, an input would be discarded if the Enter key had not been pressed.
- **Valid DisplayModels**. Only the DisplayModels that are valid for a given type of object are now offered for selection. For example, the choice of DisplayModels for a Text object is now restricted to TextModels. Formerly, all possible DisplayModels were offered.
- **ImageModels for object icons are excluded**. List of valid DisplayModels offered for selection now exclude the ImageModels for object icons. A new IconModel class has been created for these DisplayModels.
- **Menus are automatically scrolled to the present value**. Scrollable menus now scroll automatically to the present selection on the list, if applicable.
- **Improved accuracy for dragged/rotated objects**. The coordinates for Objects that are dragged or rotated through the user interface are now saved to six decimal places in the preferred units for distance and angle.
- **Main report re-sequenced**. The main report is now printed with the entities appearing in the order of palette name, class name, and entity name.
- **Input file re-sequenced**. The input file is now saved with the entities appearing in the same order as the main report. To facilitate the auditing of input files, objects and inputs that are purely graphical in nature are now printed separately at the end of the file.
- **Outputs 'PointOfInterest' and DistanceToPOI' added to View**. These outputs make it easier to select the 'DrawRange' input for an object.
- **InputBox object removed**. The 'InputBox' object has been removed from the Model Builder. This object is no longer necessary now that most numerical inputs can accept an expression, which allows an 'InputValue' object to be used. Models that already use InputBox will still work correctly.
- **DefaultEntity input removed**. The 'DefaultEntity' input for Server and many other objects in the Process Flow palette has been removed from the Input Editor. This input is no longer necessary now that the 'notNull' function can be used to write an expression that is valid when the output 'obj' is null. Models that already use the DefaultEntity input will still work correctly.
- **Width input restricted to integers**. The 'Width' input for polyline objects such as Arrow, EntityDelay, and EntityConveyor is now restricted to integer values. Formerly, it was possible to enter a decimal value that would be truncated to an integer.
- **Arrow icon**. The 'Arrow' has a new icon in the Model Builder that looks much more like the arrow that is dragged and dropped.
- **BooleanIndicator moved to the Graphics Objects palette**. 'BooleanIndicator' is used strictly for model graphics, so it was inappropriate for it to appear in the Basic Objects palette.
- **Bug fix for resizing an object**. Objects whose 'RelativeEntity' input has been specified can now be resized successfully.

- **Bug fix for Speed Multiplier**. It is now possible to select a 'Speed Multiplier' value of less than one in countries that use a comma as decimal separator.

Changes in release 2018-04:

- **Scrollable menus.** Menus that include more than 10 entries are now displayed with a scroll bar.
- **ShowEntities keyword for EntityContainer.** A new keyword 'ShowEntities' allows the entities stored in an EntityContainer to be hidden.
- **Clicking on the background.** The Input Editor and Output Viewer are now cleared when the background of a View window is clicked. Formerly, the Input Editor and Output Viewer were set to the View, but this behaviour is no longer required now that a View window can be selected from the Object Selector.
- **Bug fix for EntityConveyor.** A bug that could cause an entity to traverse the conveyor instantaneously has been fixed. The bug was introduced in release 2018-01 and only occurred when an entity was added to the conveyor at the exactly time that two or more entities had reached the end of the conveyor.
- **Bug fix for QueueLengthTimes.** A bug that prevented the QueueLengthTimes output from being shown correctly in the main report has been fixed. The bug was introduced in release 2018-02.
- **Bug fix for undoing entity deletion.** Model inputs that change when an entity is deleted are now tracked correctly by the undo/redo system.
- **Bug fix for headless mode.** A bug that prevented headless mode from working has been fixed. The bug was introduced in release 2018-02.
- **Double square brackets no longer supported for strings.** The undocumented use of double square brackets to enclose a string is no longer supported. Strings must be enclosed by double quotation marks ("). Old input files that use this format can be modified with a text editor such as Notepad++ to replace '[[' and ']]' with double quotation marks, e.g. replace [[abc]] with "abc".
- **Double quotation marks no longer supported for comments.** The undocumented use of a double quotation mark to indicate the start of a comment is no longer supported. Comments must now start with a hash character (#). Old input files that use this format can be modified with a text editor such as Notepad++ to replace the double quote mark at the start of a comment with a hash character, e.g. replace "comment with #comment.

Changes in release 2018-03:

- **Input Builder**. A new 'Input Builder' dialog box has been added to help prepare complex inputs such as the ones that accept expressions. The Input Builder includes the following features:
  - Multiple entries (enclosed by curly braces) are shown on separate lines.
  - Buttons are provided for every type of operation and function in the expression system.
  - Auto-complete is provided for both entity and output names.
  - Pop-ups are provided for all operations, functions, and outputs.
  - Input text is loaded after each keystroke, and the value of the resulting input or an error message is displayed.

  The Input Builder is launched by clicking on the arrow button in the Input Editor cell for the input to be entered. It can be used for inputs to the RunOutputList, AttributeDefinitionList, CustomOutputList, AttributeAssignmentList, and any other keyword that accepts one or more expressions.

- **RunOutputList keyword**. The RunOutputList keyword now accepts one or more expressions that return a non-dimensional quantities (non-dimensional number, entity, string, or array) without providing an input to the UnitTypeList keyword.

- **UnitTypeList keyword**. The descriptions for the UnitTypeList keyword for Simulation, ExpressionLogger, and EntityLogger, and the matching RunOutputList and DataSource keywords have been modified to suggest how to avoid using the UnitTypeList keyword. This input will be removed at some point in the future.
- **View windows**. The View windows are now shown in the Object Selector. Views can be deleted, duplicated, and renamed in the same way as other entries in the Object Selector. However, a renamed view window must be closed and re-opened before its new name is displayed.

Changes in release 2018-02:

- **Custom outputs**. Custom outputs defined using the CustomOutputList keyword can now refer to custom outputs defined for the same object or for other objects. Note that when using the Input Editor, it is necessary to load the input for a custom output (by pressing the Return key) before referencing it in another custom output.
- **TotalTime output**. A new 'TotalTime' output has been added to all entities that record state data. This output is the total of the times the entity has spent in each state. For a generated entity, this time is the total time since the entity was generated, i.e. the total system time.
- **State Statistics**. The Statistics object has been extended to automatically collect statistics on the time in each state for the entities it processes. New outputs EntityTimeMinimum, EntityTimeMaximum, EntityTimeAverage, and EntityTimeStandardDeviation provide basic statistical information for each state that is recorded from the processed entities.
- **Histogram data**. The Statistics object can now collect histogram data on the variable being tracked. New outputs HistogramBinCentres and HistogramBinFractions provide the histogram data.
- **TextHeight keyword**. A new keyword 'TextHeight' has been added to TextModel, which provides the default text height for the objects that use a particular TextModel. This feature allows a user to change the height of multiple Text objects with a single input, similar to the way that text colour and style can be changed. The TextHeight keyword for Text now appears under the 'Font' tab along with the other font-related keywords.
- **SimEntity initialisation**. A SimEntity generated by an EntityGenerator will now retain its time-in-state data at the end of the initialisation period (warm-up). This change has been made to avoid having a Statistics objects record artificially small times for some states that would become the minimum times for those states.
- **FileToMatrix and FileToVector**. Time stamp, entity name, expression, and string entries in the data file are now recognized automatically. To make it easier to find errors in the data file, any entry that cannot be interpreted correctly as a time stamp, entity name, or expression is assumed to be a string. With this change, the DataFormat keyword is no longer necessary and has been removed. The error message that results from loading an input file that still uses this keyword can be ignored. It will be corrected automatically when the model is re-saved.
- **Comments in a data file**. The data file read by a FileToMatrix or FileToVector object can now include comments. Any text in a record that appears after a '#' symbol will be ignored. This feature makes it possible to add column heading to the data file. For this purpose, it is best to use spaces to separate the entries in each record instead of tabs. Note that the data file is delimited by both spaces and tabs, and that multiple space or tab characters can be used between entries.
- **EntityGenerator names**. A new keyword 'BaseName' has been added to EntityGenerator to allow the user to set the names for the generated entities.

- **TickLength**. The upper bound of 5 seconds has been removed from the TickLength keyword for Simulation.
- **GlobalSubstreamSeed**. The random number generator has been modified to speed up its re-initialization between replications. The speed up occurs only if the GlobalSubstreamSeed is increased with each replication, which is the typical procedure.
- **NumberToStart keyword**. A new keyword 'NumberToStart' has been added to the Pack and AddTo objects. This input determines the number of entities required in the queue before beginning to pack an EntityContainer. Formerly, packing did not begin until all of the entities were present in the queue.
- **Bug fix for Seize**. A bug has been fixed that prevented a Seize object from restarting when one of its thresholds re-opened. The bug had been introduced in release 2018-01.
- **Thresholds for the Seize object**. The ImmediateThresholdList and ImmediateReleaseThresholdList keywords have been removed from the Seize object. Because resources are seized instantaneously, there is no need to distinguish between the OperatingThresholdList and the ImmediateThresholdList (or the ImmediateReleaseThresholdList).
- **Deleting a Polyline node**. A node in a polyline object such as EntityConveyor, EntityDelay, etc. can now be removed by right-clicking on the node and selecting 'Delete Node' from the context menu. This method is much easier to remember than the awkward keystroke combination Shift+Alt+Control Left-Click!
- **Line colour and width**. The line colour and width for EntityConveyor, EntityDelay, FluidPipe, and FluidFixedFlow can now use the inputs in their DisplayModel as their default setting. Separate Polyline models have been provided as the default DisplayModel input for each type of object. This allows the user to set the inputs for width and colour for all the EntityConveyors, etc. in a model at once. For consistency, the colour and width keywords for each of these object types have been moved to the Graphics tab.
- **NonStatExponentialDist validation.** The NonStatExponentialDist object now checks that its TimeSeries starts with zero expected arrivals at time zero and that the expected arrivals increase monotonically with time. These checks are performed when the model is started.
- **EntitySink restriction**. EntitySink has been prevented from deleting a dragged-and-dropped entity. Only entities generated by an EntityGenerator will be deleted.
- **VisibleViews and DrawRange keywords**. Keywords VisibleViews and DrawRange have been added to DisplayEntity (every visible object in the model). When set, these override the corresponding inputs to DisplayModel.

Changes in release 2018-01:

- **EntityProcessor**. A new 'EntityProcessor' object has been added to the Process Flow palette. This object acts as multiple parallel servers, with the additional feature that resources can be seized and released for each entity. The effective number of servers can change during the simulation run. Increases in capacity are acted on immediately, but decreases in capacity become effective only when entities are released (the same rules as the capacity of a Resource). The opening and closing of thresholds and downtime objects affects all the parallel servers simultaneously.
- **Match Value outputs for Queue**. Three new outputs, 'UniqueMatchValues', 'MatchValueCountMap', and 'MatchValueMap', have been added to Queue to provide information related to the Match values for the entities in the Queue.
- **Sizes and positions of tool windows**. The sizes and positions of the tools windows (Model Builder, Object Selector, Input Editor, etc.) are now saved as part of the input file. The width of the Control Panel is also

saved. This feature allows users to re-arrange these windows and retain the result to suit their monitor size and model layout. The default sizes and positions can be restored by selecting 'Reset positions and sizes' from the Tools dropdown menu.

- **Resetting View windows**. The default sizes and positions of the View windows can be restored by selecting 'Reset positions and sizes' from the Views dropdown menu. This feature is useful when loading a model that was prepared on a computer with a different size monitor or with multiple monitors.
- **Arrowheads for PolylineModel.** The ability to show an arrowhead has been added to PolylineModel. This change removes the need for a separate ArrowModel class, which has been deleted.
- **Degrees.** The default unit for angles has been changed from radians to degrees.
- **Splitting polylines.** A new item, 'Split' has been added to the context menu for a polyline type object, such as Arrow and EntityConveyor. When a node in the polyline object is right-clicked and 'Split' is selected, the object is divided into two separate polyline objects at the location of the node.
- **Optimisation of 'ImmediateThresholdList' and 'ImmediateReleaseThresholdList' inputs.** Models that make heavy use the ImmediateThresholdList and ImmediateReleaseThresholdList inputs should see somewhat greater execution speed.
- **'Save Changes' prompt on starting a run.** The prompt to save changes, which occurs when the Play button is pressed for an unsaved model, has been modified to be consistent with other prompts to save changes.
- **Bug fix for the 'Save' button.** Deleting or renaming an object now activates the 'Save' button and triggers a prompt to 'Save Changes' if the model is started or closed.
- **Bug fix for Batch mode.** The bug introduced in release 2017-11 that prevented Batch mode ('-b') from working has been fixed.
- **Bug fix for EntityConveyor.** An entity that is added to a stopped EntityConveyor is now shown correctly at the start of the conveyor. Formerly, the entity would not be moved graphically to a stopped conveyor.
- **Bug fix for Resource.** Only models that have multiple Seize objects that use the same Resource AND that specify inputs for the 'Match' keyword are affected.
- **Bug fix for ImmediateReleaseThresholdList keyword.** The ImmediateReleaseThresholdList feature now works correctly for EntityConveyor. Formerly, entities on the conveyor were stopped instead of being released.

Changes in release 2017-12:

- **Bug fix for multiple runs**. Corrects an error introduced in release 2017-11 that prevented multiple runs for models that use EntityContainers.

Changes in release 2017-11:

- **Event Viewer**. A new tool has been added that displays the future events and conditional events as the simulation executes. The last events that have been executed are also displayed (up to 1,000 events). Buttons are provided to execute the next event or to execute all the events until the next event time.
- **Undo/Redo buttons**. Buttons have been added to undo one or more of the last changes to a model. Buttons to redo changes have also been added.
- **Button Bar**. A button bar has been added to the Control Panel that includes new buttons for 'New', 'Open', 'Save', 'Undo', and 'Redo', and relocated buttons for '2D', 'Show Entity Flow', and 'Create Entity Links'. Options menu items for 'Show Axes', 'Show Grid', and 'Snap to Grid' have also been converted to buttons and placed on the button bar.

- **GUI keywords**. Most of the Simulation keywords under the GUI tab have been removed from the Input Editor. The easiest way to set these inputs is through the graphical user interface, so there is no need to include them in the Input Editor.

Changes in release 2017-10:

- **Match keyword generalized**. The Match keyword for Queue, Server, Seize, etc. can now accept strings and entities as well as integers.
- **Bug fix for Thresholds**. The keywords ShowWhenClosed and ShowWhenOpen now work correctly when the Threshold's DisplayModel input has been changed.

Changes in release 2017-09:

- **Bug fix for FileToVector and FileToMatrix**. An ExpressionEntity can now use the Value output for FileToVector or FileToMatrix successfully. Formerly, this caused a null pointer error to be generated when the model was loaded.
- **Bug fix for EntityConveyor**. The TravelTime keyword can now accept an expression that depends on the Value output for FileToVector and FileToMatrix. Formerly, this caused a null pointer error to be generated when the model was started.
- **Bug fix for Text objects and EntityLabel under OSX**. Unprintable characters are prevented from being added to a Text object or EntityLabel. Formerly, under OSX, pressing the Shift key when editing a Text object or EntityLabel caused an unprintable character to be added.
- **Bug fix for drag and drop under OSX**. Dragging and dropping now works correctly under OSX. Formerly, a dragged and dropped object would appear at the edge of the window instead of at the location where it was dropped.

Changes in release 2017-08:

- **FileToVector and FileToMatrix can read strings, timestamps, entities, etc.** See the User Manual for details.
- **Improved formatting for floating point numbers.** The 'format' function can use the Java formatting codes for floating point numbers. The most useful code is %.nf which displays a number with n decimal places. For example, 'format("x = %.2f", 1.234)' returns "1.23".
- **Keyword input examples.** Example inputs in keyword pop-ups now show exactly what the user can enter in the Input Editor. Formerly, they were shown in the format used in the configuration file.
- **Units can be entered with square brackets.** Users can now enter numbers with units in the Input Editor with the units enclosed by square brackets, if preferred. For example, the input 4 h could be entered as 4 [h].
- **Improved keyword pop-ups for AttributeDefinitionList and CustomOutputList.**
- **Bug fix for DowntimeEntity.** The keywords Interval and Duration can now accept expressions correctly.
- **Bug fix for Arrow.** Arrowheads now respect the RelativeEntity and Region settings for the entity.

Changes in release 2017-07:

- **indexOf function**. A new function has been added that returns the index in an array for a specified entry. For example, 'indexOf({5, -1, 2}, -1)' returns 2.
- **format function**. A new function has been added that constructs a string using a format string and one or more additional arguments. Each argument can be a number, string, object, array, or map. This function mirrors the String.format method provided in Java. Each entry of %s in the format string inserts the next

argument in the resulting output text. For example, 'format("x = %s", 5)' returns "x = 5.0". For dimensional numbers, divide by 1 in the units in which the number is to be displayed, e.g. 'format("x = %s cm", 5[m]/1[cm])' returns "x = 500.0 cm".

- **Concatenating and appending an array**. Arrays can be concatenated or appended using the '+' operator. For example, '{1, 2, 3} + {4, 5, 6}' concatenates the two arrays, returning {1, 2, 3, 4, 5, 6}. When the entry on the right hand side of the '+' operator is not an array, it is appended to the array on the left hand side, e.g. '{1, 2, 3} + "abc"' returns {1, 2, 3, "abc"}.
- **Automatic conversion to a string**. Numbers, objects, arrays, and maps can be appended to a string using the '+' operator. These quantities are converted automatically to a string prior to appending. For example, '"abc" + 5' returns "abc5.0".
- **Index argument for map and filter**. The higher-order functions map and filter now have the option of accepting a two-argument lambda function, where the second input is the index of the element in the array. For example, 'map(|x, i|(2*i), {5, 8})' returns {2, 4}, and 'filter(|x,i|(i>2), {5, 6, 7, 8})' returns {7, 8}.
- **EntityLauncher**. This new object, in the Process Flow palette, creates an entity when it is clicked by the user or when a specified keyboard key is pressed. It is intended to allow users to create interactive models or even simple games.
- **More improved keyword pop-ups**. Many keywords that require specially formatted inputs now have better descriptions of this format in their pop-ups.
- **Bug fix for Duplicate**. Entities created by the Duplicate object now reproduce the present graphics of the source entity.
- **Bug fix for Windows 10**. View windows created under Windows 10 now retain their correct size and position on the computer screen when the model is reloaded.

Changes in release 2017-06:

- **Sort function**. A higher-order function that applies a two-input lambda function to the elements of an array and returns an array that has been re-ordered so that the lambda function returns TRUE (i.e. a non-zero number) for each adjacent pair of elements. The lambda function should return FALSE for entries that are equal. For example, 'sort(|x, y|(x>y), {2,3,1})' returns {3, 2, 1}. With an appropriate lambda function, the array to be sorted can contain objects or other types of entries.
- **Range function**. A function that generates an array of numerical values that can be used as an input to the higher-order functions. The function can have one, two, or three arguments, e.g. 'range(3)' returns {1,2,3}, 'range(2,4)' returns {2,3,4}, and 'range(2, 3, 0.5)' returns {2.0, 2.5, 3.0}.
- **Local variables**. It is now possible to define one or more local variables within an expression. A local variable can be defined for any valid type, i.e. a number, string, object, array, map, or lambda function. The following syntax is used: '<variable1> = <expression1>; … <variableN> = <expressionN>; <final expression>'. For example, the expression 'x = 1; y = 2; x + y' would return the value 3.
- **Data files**. Two new objects, FileToVector and FileToMatrix, have been added to the Basic Objects palette that read numerical data contained in a file. The data must be delimited by either spaces or tabs (but NOT commas). Each of the two objects make the data available to the model through an output named "Value". For FileToVector, the Value output returns a single array of data that combines all the records in the file. For FileToMatrix, the Value output returns an array of arrays, with one internal array for each record that was read. The data file is first read when the model is loaded. It is re-read and the Value output updated whenever the object receives an entity.
- **Test for probability distribution inputs MinValue and MaxValue**. An error message is now generated if a probability distribution function cannot produce a value between the MinValue and MaxValue inputs

after 1,000 attempted samples. Formerly, this condition would cause the program to stall with an infinite loop.

- **EntityList output for EntityContainer**. A new output has been added to EntityContainer that returns an array containing the entities held by the container.
- **SeizedUnits output for Seize**. A new output has been added to Seize that returns the number of units for each resource that was seized by the most recent entity.
- **AvailableUnits output for Resource**. A new output has been added to Resource that returns the number of units that are not seized at the present time.
- **Keyword pop-ups**. Keywords that can accept an expression now provide this information more clearly in their pop-ups.
- **Bug fix for SetGraphics keyword TargetEntity**. The default value for the TargetEntity keyword now works correctly. The previous release had broken this feature.

Changes in release 2017-05:

- **Lambda functions**. A lambda function is an expression that takes one or more input variables and returns a number, string, object, array, map, or another lambda function. Lambda functions can be used with higher-order functions to perform complex calculations that would otherwise require a loop structure (see below). A lambda function can be defined as follows. Input variables are enclosed by bars and separated by commas. The expression that generates the returned value is enclosed by brackets. For example, the lambda function |x,y|(x+y) sums two inputs. Input variables can be a number, string, array, map, or another lambda function. The object returned can be any of these same types of objects. A lambda function can be evaluated by providing inputs in the normal fashion, e.g. |x,y|(x+y)(1,2) returns 3.
- **Map function**. A higher-order function that applies a one-input lambda function to each element of an array and returns an array with the resulting values. For example, map(|x|(2*x), {1, 2}) returns {2, 4}.
- **Filter function**. A higher-order function that applies a one-input lambda function to each element of an array and returns an array with only the ones that return TRUE (i.e. a non-zero number). For example, filter(|x|(x>2), {1, 2, 3, 4}) returns {3, 4}.
- **Reduce function**. A higher-order function that applies the first input of a two-input lambda function to each element of an array. The second input to the reduce function is the initial value to be assigned to an internal value maintained by the function during the calculation. The result of the calculation for each element is assigned to this internal value. After the last element is processed, the internal value is returned. The reduce function has three inputs: the function, the initial value to be assigned to the internal value, and the array to be processed. For example, reduce(|x, accum|(x + accum), 0, {1,2,3}) returns 6.
- **MimicEntity**. A new object has been added to the Graphics Objects palette that changes its appearance to match the object specified by its SourceEntity keyword. This keyword can accept either a fixed object or an expression that return an object.
- **StateGraphics keyword for SimEntity**. A SimEntity can now change its appearance depending on its present state, using its new StateGraphics keyword.
- **TargetEntity keyword for SetGraphics can accept an expression**. The TargetEntity keyword for SetGraphics can now accept either an object or an expression that returns an object.
- **Snap-to-Grid refinements**. When an object is dragged horizontally, only its x- and y-coordinates are snapped to the nearest grid point. Its z-coordinate is left unchanged. Similarly, when an object is dragged vertically, only its z-coordinate is snapped to the nearest grid point.

- **New outputs for Server and EntityGenerator**. Server and EntityGenerator have new outputs that allow the processing/generation of an entity to be tracked.
- **Bug fix for displaying an array or map in a Text object**. Text object can now display an array or a map. This ability had been lost in an earlier release.
- **Bug fix for displaying an array of numbers**. An array containing a numbers with units is now displayed correctly when the displayed unit differs from the default unit (i.e. the SI unit).

Changes in release 2017-04:

- **Bug fix for Linux computers**. Input files are now saved correctly when running JaamSim under Linux.
- **Bug fix for EntityLogger and ExpressionLogger**. A null pointer error is no longer generated when a run is resumed beyond the end of the normal run time set by RunDuration.

Changes in release 2017-03:

- **Unit selector**. A new entry labelled 'Units' has been added the Control Panel's menu bar. This tool will allow the user to select the units in which to display model outputs in the Control Panel, Output Viewer, and output reports. For example, if you want to display outputs with units of time in seconds instead of hours, select 'TimeUnit' from the drop down menu and click on the entry labelled 's'. With this change, the Control Panel will display simulation time in seconds and every output in the Output Viewer with units of time will be displayed in seconds.
- **Windows 10**. The ugly gaps between the various JaamSim windows that appeared under Windows 10 have been eliminated.
- **Multiple prototype entities for EntityGenerator**. It is now possible to generate various different types of entities using a single EntityGenerator. The PrototypeEntity keyword can now accept an expression that returns an entity. The expression is evaluated every time a new entity is to be created. At present, this is the only keyword where an entity input can be replaced by an expression that returns an entity.
- **Generalized 'choose' function**. The 'choose' function has been extended to handle all types of expression results (i.e. numbers, strings, objects, arrays, and maps). The only restriction is that all the choices must be the same type, e.g. all numbers, all strings, etc.
- **Circular arc**. A new 'CIRCULAR_ARC' option has been added under the CurveType keyword for objects that use a polyline type DisplayModel, such as EntityDelay and EntityConveyor. When this option is selected, the object is shown as a smooth curve with constant curvature that passes through the first, second, and last points. Any additional points are ignored by the curve.
- **Snap-to-grid enforced when moving an entire polyline**. When snap-to-grid is selected and an entire polyline is dragged, the first point of the polyline is constrained to land on a grid point. Formerly, the snap-to-grid setting was ignored when an entire polyline was moved.
- **Bug fixes for expressions**. Several bugs were fixed relating to creating an array of entities within an expression.
- **Bug fix for arrowheads**. A bug has been fixed that had allowed an arrowhead to become detached from its polyline.
- **Bug fix for setting Region and RelativeEntity for a polyline**. A bug has been fixed related to setting Region and RelativeEntity for a polyline using the context menu (right click).

Changes in release 2017-02:

- **Resource responds immediately to an increase in capacity**. A Resource will now take immediate advantage of an increase in capacity allowing waiting entities to be processed. Formerly, a capacity increase was not recognized until an entity arrived at a Seize or Release object for that Resource.
- **TimeSeries can be used in an ExpressionThreshold**. A TimeSeries now executes an event whenever its value changes, allowing it to be used successfully in an ExpressionThreshold. Formerly, an ExpressionThreshold would not respond to a change in a TimeSeries value until some other object triggered an event.
- **Links to/from EntityLogger**. Arrows showing the flow of entities to and from EntityLogger have been restored.
- **Default unit types for outputs**. All objects that generate model outputs now assume DimensionlessUnit as their default unit type. EntityLogger, ExpressionLogger, Graph, and the run output report for Simulation have been modified in this way. The Text object already uses DimensionlessUnit as its default.

Changes in release 2017-01:

- **Strings in expressions.** String literals in an expression can now be entered using double quotes instead of double square brackets, i.e. "text" instead of [[text]]. Expressions that use this new format MUST be enclosed between single quotes, regardless of whether they contain any spaces. Double square brackets will continue to be supported for a few months to provide backward compatibility.
- **New comment character for input files.** The comment character for input files has been changed from double quote (") to the hash symbol (#). Input files that use the double quote symbol for comments will continue to be supported.
- **EntityLogger reformatted.** The EntityLogger object has been modified to be consistent with the ExpressionLogger object. Users can now specify which of the received entities' outputs should be logged. The resulting values are now saved in tabular format in the log file, reducing the file size and making it easier to analyse in a spreadsheet.
- **Pop-up messages for 3D loader errors.** A 3D file that does not load successfully will now generate one or more pop-up error messages. Formerly, these messages were printed to the log file without a pop-up.
- **Folder locations retained for 3D and image files.** The locations of the last 3D and image files that were loaded are now retained between launches of JaamSim.
- **Validation check for RunDuration input.** With the default value for TickLength (1 microsecond), the longest run duration is 2^63 - 1 = 9.22 x 10^18 microseconds = 292,471 years. The input for RunDuration is now checked when a run is first started to ensure that this value is not exceeded. Runs longer than this value can be executed by entering a larger value for TickLength.
- **Progress on an API for JaamSim.** Work has begun on an application programming interface (API) that will allow JaamSim to be embedded in other programs and allow these programs to launch multiple runs on separate processor cores.

Changes in release 2016-24:

- **Bug fix for headless mode**. A bug introduced in release 2016-23 that prevented headless mode from working has been fixed.

Changes in release 2016-23:

- **Create Entity Links button**. A new button has been added to the graphical user interface to allow Process Flow objects to be linked together. After clicking on the button to select this mode, the user can click on a

series of objects in succession, following the intended flow of entities, to set each object's NextComponent input. The linking process can be interrupted by clicking on the background or on an object that does not support this function. The same procedure can be used to set the PrototypeEntity input for an EntityGenerator and the WaitQueue input for a Server (or any other object that uses this keyword). The NextComponentList input for a Branch object can be set by adding each destination one at a time. It is best to select the "Show Entity Flow" button when using this tool.

- **2D/3D modes**. The 2D button has been modified to toggle between 2D and 3D modes, and the 3D button has been removed. In 2D mode, the view is locked to prevent the user from accidentally rotating the view. The 2D/3D mode is set individually for each view window.
- **New "notNull" function**. This new function tests whether an entity exists (i.e. is not null). For example, if a custom output is added to a Server that returns the value of an attribute "attrib" for the last entity received, a potential error can be avoided by defining the custom output as follows: 'notNull(this.obj) ? this.obj.attrib : 0'
- **Bug fix for array-valued attributes**. The bug reported by Zoli related to the use of a constant array in an Assign object has been corrected.
- **Bug fix for custom outputs**. The bug reported by Zoli related to the use of "this" in a custom output has been corrected.
- **Bug fix for RenegeTime**. The logic for reneging from a queue has been modified to operate efficiently when a very large RenegeTime input has been specified.
- **Bug fixes for importing 3D content**. The importers for DAE and OBJ files have been corrected.
- **Improved error messages**. Further improvements have been made to the error messages for expressions.

Changes in release 2016-22:

- **Entity flow arrows.** A button has been added to the graphical user interface that allows the flow of entities between objects to be displayed. For example, if an EntityGenerator passes its entities to a Server, then an arrow will be shown from the EntityGenerator to the Server.
- **Improved run-time error messages for expressions.** Run-time error messages involving expressions now show the location of the error within the expression.
- **New positions for the Property Viewer and Log Viewer.** The Property Viewer and Log Viewer are now shown on the right hand side of the monitor.
- **Keyword and output pop-ups**. Numerous corrections and improvements have been made to the pop-up descriptions for keywords and outputs.
- **Bug fix for ExitAtPauseCondition.** The ExitAtPauseCondition keyword now works correctly when multiple runs are specified. The meaning of this keyword and the way it works with the ExitAtStop keyword has been adjusted somewhat. See the keyword pop-up for details.
- **Bug fix for Seize.** The Seize object can now remove multiple entities from its queue, one after another, when a Match condition has been specified.

Changes in release 2016-21:

- **Array-valued attributes**. An attribute can now be created with an array as its value. The elements of the array can be numbers (with or without units), strings, entities, or other arrays. For example, the following entry to the AttributeDefinitionList input creates an array-valued attribute "attrib" whose elements are 5 seconds, 6 seconds, and 7 seconds: { attrib '{ 5[s], 6[s], 7[s] }' }. Note that the curly braces indicating an array must appear within single quote marks.

- **Size and indexOfNearest functions**. Two new functions have created for working with array and maps. The size function returns the number of elements in the array or map. For example, the expression 'size({ 5[s], 6[s], 7[s] })' would return the number 3. The indexOfNearest function returns the index of the element in an array of numbers that is closest to the specified value. For example, the expression 'indexOfNearest({ 7, 4, 12 }, 3)' would return the number 2.
- **Font keywords added to Text**. The keywords FontName, FontColour, FontStyle, DropShadow, DropShadowColour, and DropShadowOffset have been added to the Text object. These inputs allow the user to format an individual Text object by overriding the corresponding inputs for the TextModel. Formerly, a user would have to create a new TextModel every time a new format was needed. These keywords are available for all the text objects: Text, OverlayText, BillboardText, and OverlayClock.
- **RetainAll keyword added to Combine**. Normally, a Combine object passes only the entity from the first queue to the next component, and destroys the matching entities from the other queues. The new RetainAll keyword, when set to TRUE, instructs the Combine object to preserve all the matching entities and to pass them simultaneously to the next component. For users who are familiar with Arena, this behaviour is similar to the MATCH object in that software.
- **QueueList output added to Queue**. The new QueueList output returns an array containing the entities in the queue. For example, QueueList(1) is the entity at the front of the queue.
- **TimeAverage and TimeStandardDeviation outputs added to Statistics**. The new TimeAverage and TimeStandardDeviation outputs return the weighted average and standard deviation of the recorded value, where the weight is equal to the duration of a value before it is overwritten by the next value.
- **OpenCount and ClosedCount outputs added to Thresholds**. The new OpenCount and ClosedCount outputs return the number of times that a threshold has changed from closed to open and from open to closed, respectively. These outputs are available for all the thresholds objects: ExpressionThreshold, TimeSeriesThreshold, and SignalThreshold.

Changes in release 2016-20:

- **Min and Max of a collection**. Four new functions have been added to the expression system: minCol, maxCol, indexOfMaxCol, and indexOfMinCol. The argument for these functions is a "collection", that is, an indexed array or a "map" that connects keys with values. The StateTimes output for Server is an example of a map output. The maxCol and minCol functions return the maximum and minimum values in a collection, while indexOfMaxCol and indexOfMinCol return the index or key that corresponds to the maximum or minimum value. For example, if the Idle state for Server1 has the most time accumulated, then indexOfMaxCol([Server1].StateTimes) would return the string [[Idle]].
- **Bug fix for BarGauge**. An input of NaN to DataSource is now interpreted as zero. Formerly, this input caused logical problems for the graphics system.
- **Bug fix for Combine and Assemble**. The outputs NumberAdded and NumberInProgress now work correctly for Combine and Assemble. Formerly, each entity received by these objects was counted towards the NumberAdded, while only the combined or assembled entity was counted towards the NumberProcessed. This discrepancy caused the NumberInProgress output to grow without bound. Now, a set of entities to be combined or assembled count as one towards the NumberAdded.
- **Bug fix for EntityGenerator**. The output obj is now set to the last entity that was generated.

Changes in release 2016-19:

- **"Clearing_while_Stopped" state eliminated**. Formerly, when an OperatingThreshold input closed a Server (or another object that use this keyword), the Server's state would be set to

"Clearing_while_Stopped" while an entity being serviced was completed. With this change, the Server remains in the Working state while an entity is completed.

- **BarGauge**. This new object in the Graphics Objects palette allows an output value to be displayed graphically.
- **Bug fix for Text**. Formerly, the DataSource keyword for Text would incorrectly report an input error when a simple expression of the form [Entity1].Output1 was entered.
- **Bug fix for FluidTank**. The FluidTank object is now displayed correctly when it is first dragged and dropped. Formerly, it was invisible until the model was started.
- **Preparations for array valued expressions and attributes**. Work is underway to extend the expression system and attributes to handle arrays of numbers, entities, and strings. The present release includes significant changes that will permit these new features to be implemented. Please report any bugs that have been introduced.

Changes in release 2016-18:

- **Bug fix for custom outputs**. The values for custom outputs are now reported correctly in the output report.
- **Bug fix for deleted objects.** After deletion, an object has no lingering effect on the behaviour of other objects. Note however, that it is still the user's responsibility to correct any expressions that referenced the deleted object.
- **Downtime can start while a threshold is closed.** A downtime or maintenance activity can now begin when a threshold is closed.

Changes in release 2016-17:

- **Expression system extension to strings and entities**. Expressions can now return a string or an entity in addition to a number. The existing operators equal to (==) and not equal to (!=) have been extended to compare strings and entities. The plus operator (+) has been extended to concatenate two strings. Within an expression, a string constant must be enclosed by two pairs of square brackets, e.g. [[abc]] represents the string 'abc'.
- **Keyed outputs in expressions**. Keyed outputs such as StateTimes can now be referenced within an expression. For example, the expression '[Server1].StateTimes([[Idle]])' will return the total time that Server1 has been in the 'Idle' state.
- **String- and entity-valued attributes**. Attributes have been extended so that they can store a string or an entity as well as a number. The Assign object has been modified to allow attributes to be assigned these types of values. The initial value for a string- or entity-valued attribute is assigned by using an expression in the AttributionDefinitionList input. For example, the input { attrib1 [[abc]] } would define attribute attrib1 to be string-valued with the initial value 'abc', while the input { attrib2 [Server1] } would define attribute attrib2 to be entity-valued with the initial value Server1.
- **Smooth curves**. Linear objects such as Arrow, EntityConveyor, etc. can now be configured as smooth curves instead of polylines. The new keyword CurveType under the Graphics tab allows the user to choose between Linear, Bezier, and Spine.
- **Bug fix for PriorityValues and MatchValues outputs**. The Queue outputs PriorityValues and MatchValues can now be used in expressions.

Changes in release 2016-16:

- **Major code changes to Server, EntityConveyor, Pack, Unpack, etc**. These modifications should have no effect on model results. Please be on the lookout for any unexpected changes.
- **Bug fix for EntityConveyor**. A bug has been fixed that was related to starting or stopping a conveyor at the same instant that an entity is received or released.
- **Faster response for pop-ups**. Pop-ups for input and output descriptions, etc. have been made more responsive and reliable.
- **Bug fix for expressions that use the % operator**. Input error messages were being corrupted for expressions that contained the % operator.
- **Release object can free zero units of a resource**. The Release object can now accept an input of zero for the NumberOfUnits keyword.

Changes in release 2016-15:

- **An input of zero is allowed for the Pack keyword NumberOfEntities**. Formerly, a value of zero caused an input error message. More troublesome was an expression that returned zero part way through a simulation run, causing a run-time error. The Pack object now assumes a minimum value of one entity per EntityContainer, even if zero is entered. Note that allowing zero entities per EntityContainer would cause an infinite number of containers to be generated instantly.
- **An input of zero is allowed for the AddTo keyword NumberOfEntities**. For the AddTo object, an input of zero for NumberOfEntities causes the EntityContainer to be release immediately with no entities added.
- **An input of zero is allowed for the ReleaseFrom keyword NumberOfEntities**. An input of zero for NumberOfEntities causes the EntityContainer to be release immediately with no entities removed.
- **ValuePrecisionList keyword added to ExpressionLogger**. The new keyword allows users to specify the number of decimal places to display for each entry in the report generated by the ExpressionLogger object.
- **Bug fix for EntityConveyor related to ImmediateThresholdList**. EntityConveyor can now handle the situation where an entity is added to the conveyor at the same instant that the threshold opens.
- **Modified the default XAxisLabelFormat for Graph**. Previously, the text "s" was appended to the labels on the x-axis to indicate the default unit of "seconds". This default was unnecessary and most users found it to be confusing.
- **Output Viewer shows vector-valued outputs in selected units**. Formerly, the outputs Position and Size were not converted to the units specified by the DisplayedUnits input for the Simulation object.
- **Snap-to-Grid set by default**. The Snap-to-Grid setting for new models has been set to TRUE. This setting makes it much easier to align the objects in a model and should be used for almost all models.
- **WorkingStateList keyword has been removed from objects where it is not required**.
- **Faster killing of entities in large models**. Generated entities can be deleted much more efficiently from models with large numbers of entities (more than 10,000).

Changes in release 2016-14:

- **ImmediateReleaseThreshold keyword added to Server, etc**. The behaviour of the new keyword is similar to the ImmediateThresholdList, except that the entity being process by the Server, etc. is released immediately instead of waiting for processing to be resumed.
- **Improved importing of 3d assets from SketchUp**. Many more of the Collada files in the 3d Warehouse can now be imported successfully.

- **Object Selector shows entities in natural order**. The Object Selector has been modified to display the entities in natural order instead of order of definition.
- **Object Selector protected from excessive entities**. An upper limit of 10,000 has been placed on the total number of generated entities shown in the Object Selector. Objects defined by the user are not affected by this restriction. Only the temporary entities created by the EntityGenerator or similar objects are restricted. Formerly, the Object Selector would cause the graphical user interface to lock up when large numbers of generated entities were present.
- **Graph speedup**. The Graph object has been optimized to reduce the time required to record data. Many models that include one or more graphs will run significantly faster.
- **Example for coding new objects**. A new object "DemoEntity" and palette "Examples" has been provided as a simple example of how to program new objects and palettes for JaamSim. By default, the Examples palette and DemoEntity are not shown in the Model Builder, however, they can be activated by editing the file "resources/inputs/examples.inc" and deleting the line of input indicated by a comment.

Changes in release 2016-13:

- **Headless mode for running on a server**. A new command line tag -h (or -headless) has been added that allows JaamSim to be executed on a server that has no graphics capability. The new tag activates batch mode automatically -- there is no need to include the -b tag in addition to the -h tag. Headless mode can be used with both the jar file and the executables.
- **Bug fix for Queue**. The MaxPerLine keyword now displays the correct number of queued entities in each row of the queue.

Changes in release 2016-12:

- **Unit type checking for expressions**. An expression that returns the wrong unit type will generate an error message when it is first entered in the Input Editor. Formerly, these errors were not detected until the simulation was started.
- **RunOutputList errors printed to the output file.** An error message generated by one of the expressions in the RunOutputList input will be printed to the output file.
- **Invalid inputs shown in red.** Inputs that are rendered invalid by a change to a UnitType input will be shown in red in the Input Editor instead of being cleared.
- **Validation errors printed to standard error.** Errors that are detected when a run first starts are now recorded correctly when running in batch mode.
- **Bug fix to the RenegeCondition keyword for Queue.** RenegeCondition accepts now accepts values in the range 0 - 1.
- **EntityGenerator keyword MaxNumber accepts an expression.**
- **InitialOpenValue keyword added to ExpressionThreshold.** Sets the ExpressionThreshold's state at the start of the simulation run. This keyword is needed only when the CloseCondition input is used and both the OpenCondition and CloseCondition return false at the start of the run. Otherwise, the initial state for the ExpressionThreshold is determined explicitly by its OpenCondition and CloseCondition inputs.
- **Bug fix to ExpressionThreshold.** Situations where the OpenCondition and CloseCondition both return false are handled correctly. In this situation, the state of the ExpressionThreshold should be left unchanged. Formerly, this feature did not work reliably.

Changes in release 2016-11:

- **Variable speed conveyors**. The EntityConveyor keyword TravelTime now accepts an expression, TimeSeries, etc. Note that changes to the travel time are only recognised by the model at event times for the conveyor, that is, when an entity is first added to the conveyor or when an entity reaches the end of the conveyor.
- **Bug fix for VideoRecorder**. The transparent region around an object icon and other transparent features are now recorded correctly.

Changes in release 2016-10:

- **Maintenance and breakdowns for Server, etc**. Multiple breakdown and maintenance activities can be specified for many objects in the Process Flow palette: Assemble, Combine, EntityConveyor, EntityGate, EntityGenerator, Pack, Server, and Unpack. The timing of breakdown and maintenance events can be based on either calendar time or the working time for a specified entity. A breakdown or scheduled maintenance event can follow several types of protocols: stop processing can immediately ("Immediate" type), or after the entity being processed is finished ("Forced" type), or when there are no entities waiting for processing ("Opportunistic" type). The same downtime event can affect multiple objects by using the same DowntimeEntity with each object.
- **ImmediateThresholdList**. A new keyword "ImmediateThresholdList" specifies thresholds that stop processing immediately, interrupting any work that is already in progress.
- **BooleanIndicator**. A new object called BooleanIndicator has been added to the Basic Objects palette. It displays a green or red circle depending on whether a specified expression returns true or false, respectively. It can be used with the DowntimeEntity output WorkingState to display the occurrence of downtime events.
- **Minor fixes**. Improved descriptions for TimeSeriesThreshold keywords. ReleaseDelay keyword for EntityGate now accepts an expression. Assemble now sets the state of the assembled part. Minimum and maximum value tests on inputs have been relaxed when an ExpressionEntity is entered.

Changes in release 2016-09:

- **Object name changes are tracked for Assign object**. Inputs to the AttributeAssignmentList keyword for an Assign object are now updated automatically when an object name is changed.
- **Probability Distribution objects accept expressions**. Expressions can now be as inputs to the keywords for the various probability distributions objects.
- **Value output added to BooleanSelector**. The BooleanSelector object now has an output that reports the last Boolean value that was selected.
- **New icon and splash screen**. See the JaamSim blog for more information.

Changes in release 2016-08:

- **Object name changes are tracked in expressions**. When an object's name is changed, any expression that refers to that object is updated to use the new name. The only place where this feature does not work is in the Assign object. It will be fixed in the next release.
- **Error location for expressions**. If an expression is entered incorrectly, the pop-up error message now shows the location of the error in the expression.
- **Expression bug fix**. A bug was introduced in release 2016-07 that caused serious problems when an object was deleted while it was still being referenced in an expression. This bug has now been fixed.

- **Run Output report bug fix**. The run output report generated by the RunOutputList keyword was being generated for all models regardless of whether the RunOutputList keyword was used or not. The report is now generated only when an input is provided to the RunOutputList keyword.

Changes in release 2016-07:

- **Faster execution speed.** Expression evaluation has been optimized to increase model execution speed. For models with many complex expressions and especially ExpressionThreshold objects, the increase in execution speed can be as much as 50%.
- **Selected run outputs.** New keywords RunOutputList and UnitTypeList have been added to Simulation to specify the outputs to be saved for each simulation run. These outputs are saved in column format to a new output file named <configuration file name>.dat. If the new tag "-s" is included on the command line, the run outputs will be returned to the command line (standard out) – see the next item below. When specified, run outputs will be generated regardless of the PrintReport keyword. A
- **Command line inputs and outputs**. Inputs and outputs can be piped on the command line to/from JaamSim using the following syntax:

```
program1.exe | java -jar jaamsim2016-07.jar -s -b config.cfg | program2.exe
```

  This command will instruct JaamSim to load the configuration file "config.cfg" and then process any additional configuration file inputs from program1. The run outputs specified by the RunOutputList keyword (see previous item) will be returned to the command line and processed by program2. The new tag "-s" (or "-script") tells JaamSim to read the additional inputs from the command line (standard in) and return the specified run outputs to the command line (standard out). The existing tag "-b" (or "-batch") is needed to tell JaamSim to execute the run automatically and to close itself when finished. Note that this feature only works with the .jar file, not the executable, and that it is necessary to include "java -jar", otherwise the input and output streams will not be connected properly to standard in and standard out. This feature can be demonstrated by entering:

```
echo Define DisplayEntity { Fred } | java -jar jaamsim2016-07.jar -s config.cfg
```

  This input will cause a new DisplayEntity named "Fred" to be added to the model defined by config.cfg. The "-b" has been removed from the command so that the model does not start immediately.
- **Resource logic modified to avoid deadlocks.** A new keyword StrictOrder has been added to Resource to help users avoid logical deadlocks in their models. When a resource becomes available, it is assigned to the entity that has the highest priority in its Queue. In the case of entities with equal priorities, the resource is assigned to the one that has waited the longest. If the selected entity cannot accept seize the resources because of an OperatingThreshold restriction or because it needs to seize other resources which are not available yet, then there are two options: 1) the resource can be held for the selected entity, or 2) it can be assigned to the first entity that is able to seize the resource. The StrictOrder keyword determines which option is used – TRUE for option 1 and FALSE for option 2. The default value is set to FALSE because this option is less likely to cause a deadlock. **THIS DEFAULT COULD CAUSE MODEL RESULTS TO CHANGE** – Earlier releases of JaamSim had assumed the equivalent of TRUE for StrictOrder. If this change affects the results for your model, the solution is to set StrictOrder to TRUE for all the Resources.
- **StateGraphics keyword provided for EntityDelay.** EntityDelay objects can now have their appearance change when their state changes, in the same way as Server and other objects that have the state output.
- **Bug fixes for Seize and Release.** The inputs for the ResourceList and NumberOfUnitsList keywords are now checked to ensure that the same number of entries are specified in each list.

- **Bug fixes for Input Editor related to keywords accepting a list of objects.** When specifying a list of entities for keywords such as the Graph keyword DataSource, it is necessary to include braces around each object in the list. These braces are now included correctly when the input is selected from the drop down menu in the Input Editor.

Changes in release 2016-06:

- **Attributes and Custom Outputs displayed in the Output Viewer in order of definition.**
- **"Add Label" modified.** The context menu item "Add Label" has been converted to a check box and renamed "Show Label".
- **Speed Multipliers less than 1.0 are accepted.** Previously, simulation models were restricted to run faster than real time. Now, they can run either faster or slower.
- **New names for entities created by Pack and Assemble.** Entities generated by Pack and Assemble objects are now named after these objects instead of the prototype. This is the same convention as EntityGenerator and is used to avoid multiple generated entities with the same name.
- **PauseCondition and GlobalSeed keyword bug fixes.** These Simulation keywords can now use "this" in an input expression.
- **Unnecessary error message suppressed.** An annoying error message is avoided when the Simulation object is clicked in the Object Selector, which accidentally puts it in name change mode.
- **Imported images retain their aspect ratio.** Images imported used File > Import are sized to preserve the aspect ratio of the original.
- **Sub-menu for File > Import.** Clicking on File > Import now brings up a sub-menu containing "Images" and "3D Assets". The old system of allow both images and 3D assets to be imported from the same file chooser dialog box was confusing to new users.
- **Change Graphics dialog can import images.** The Import function on the Change Graphics dialog box can now import images (i.e. pictures such as a .jpg file) as well as 3D assets.
- **Misc. improvements to the Change Graphics dialog.** DisplayModels listed in alphabetical order, only ImageModels and ColladaModels included in the list, "Cancel" button added, and layout improved.

Changes in release 2016-05:

- **Re-edit option for the Input Editor.** The Input Editor now provides an option to revise a bad input instead of always clearing it.
- **New unit choices for RateUnit.** Units of /min, /ms, /us, and /ns have been added.
- **Context menu items for RelativeEntity and Region.** The RelativeEntity and Region inputs for an object can now be selected by right-clicking on the object. The Position input for the entity is adjusted automatically so that it retains the same position in the view window.
- **CloseCondition keyword added to ExpressionThreshold.** Separate conditions can be provided to open and close an ExpressionThreshold. For example, an ExpressionThreshold can now be set to open when some quantity exceeds a value A, but only close when it falls below a different value B.
- **Menu items listed in "Natural Order".** Normally, computer programs will list the objects Queue1, Queue2, and Queue10 in the order {Queue1, Queue10, Queue2} . Menu items are now listed in the same sequence that a human would expect , i.e. {Queue1, Queue2, Queue10}.
- **Contents of an EntityContainer assigned the same state.** When an entity is added to an EntityContainer, it is now assigned the same state as the EntityContainer. Furthermore, when the EntityContainer changes state, its contents are also changed to that state.

- **Major revisions to Calculation Objects.** Output values are calculated on demand, output unit type for Integrator and Differentiator are determined automatically from the input unit type, and a new object "UnitDelay" introduced to prevent errors in a feedback loop.

Changes in release 2016-04:

- **Windows positioned correctly in Mac OS X.** JaamSim windows are now sized and positioned correctly on Mac OS X computers.
- **Reduced tolerance for Mouse Handles.** The mouse handles used to resize and rotate an object have been adjusted so that the cursor must be closer to the handle before it is activated. This change will make is harder for a user to accidentally resize an entity when he/she only intended to reposition it.
- **Resources choose between Seize object requests based on priority.** Resources now use both priority and waiting time to choose the next entity to seize Resource units. For example, if two entities are waiting in separate Queues for two Seize objects, the Resource will choose the entity assigned the highest priority (lowest numerical value) by its Queue. If the two priorities are equal, then the entity with the longest waiting time will be chosen.
- **Bug fixes related to CustomOutputList.** The spelling of the CustomOutputList keyword has been corrected. Configuration files saved under release 2016-03 should be edited to correct the spelling. A second problem related to the order of the inputs in a saved configuration file has also been corrected. However, users should avoid defining a custom output for one object using a custom output for another object.

Changes in release 2016-03:

- **User-defined Outputs**. Users can define new outputs for an object by specifying an expression to be evaluated using the CustomOutputList keyword.
- **Reneging from Queues**. Entities can renege from a Queue using the new keywords RenegeTime, RenegeCondition, and RenegeDestination.
- **Support for Windows Task Bar on left of monitor**. JaamSim windows are now positioned correctly under Windows when the Task Bar position is set to the left of the monitor.
- **Improved error handling for Text object**. An error message has been added to notify users when the UnitType keyword for a Text object must be specified. This is the case when an expression or object returning a number is entered for the DataSource keyword. It is not mandatory for a DataSource entry that returns text such as an entity's State output.
- **Pack keyword NumberOfEntities must be greater than zero**. An error message has been added to ensure that the NumberOfEntities input to the Pack object is greater than or equal to one.

Changes in release 2016-02:

- **VideoRecorder**. Keyword descriptions are improved, video file is saved to the same directory as the output report, and a warning message is shown if computer does not support video capture.
- **Control Panel size and position corrected**. A small correction has been made to the size and position of the Control Panel for computers running Windows 7 (Aero theme), Windows 8, and Windows 10. Windows 7 (Classic theme) was always shown correctly. The new borderless windows for Windows 10 are still not quite right. This is a Java problem that hopefully will be solved in later releases of the Java Runtime Environment.

- **TestEntity renamed as DefaultEntity**. The TestEntity keyword for Process Flow objects has been renamed as DefaultEntity to better describe its function. The old name is still accepted as a synonym to maintain backwards compatibility. The description of the keyword has also been revised.
- **Ceil, floor, signum, sqrt, and cbrt added**. Expressions can now include the mathematical functions for ceiling, floor, signum, square root, and cube root.
- **Improved descriptions for Process Flow objects**.
- **Output Viewer improvements**. Outputs that are empty or null are now shown explicitly in the Output Viewer. Formerly, they were shown as blank entries.

Changes in release 2016-01:

- **Transparency enabled for 2D shapes**. Flat objects that use ShapeModel for their graphics, can now be set to a transparent colour. Use the alpha or transparency setting in the colour selector tool.
- **Expressions can use indexed arrays**. Expressions can now include references to indexed arrays. This feature allows the Simulation output "RunIndex" to be used to execute multiple runs that test a range of values for one or more inputs. For example, setting the ServiceTime input for a Server to the expression '1[s] + 0.1[s]*[Simulation].RunIndex(2)' would test service time inputs of 1.1 s, 1.2 s, 1.3 s, etc., as the second run index is incremented over multiple runs. See the release note for 2015-46 under "Experimenter" for more information.
- **Polyline bug fixed**. It is now safe to right click on a polyline object and select "change graphics".
- **EntityLogger bug fixed**. The full log report is now generated by EntityLogger. Previously, the log report would sometimes end abruptly.
- **EntityLogger and ExpressionLogger can record multiple runs**. When multiple runs are performed, the results for each run are appended to the ends of these log reports.

Changes in release 2015-50:

- **NumberAdded and NumberProcessed outputs corrected for initialization period.** These two outputs for Process Flow objects are now reset to zero at the end of the initialization period.
- **NumberInProgress output added.** This output has been added to the Process Flow objects. Now that the NumberAdded and NumberProcessed outputs are reset to zero at the end of initialization, the number of entities currently being processed by an object cannot be calculated reliably by NumberAdded - NumberProcessed. For example, if one object was being processed at the end of initialization, this calculation will return NumberInProgress - 1.

Changes in release 2015-49:

- **State outputs displayed correctly prior to start of run and on reset**. The state outputs: State, WorkingState, and WorkingTime are now shown correctly prior to the start of a run and when a run is reset.

Changes in release 2015-48:

- **New configuration file syntax "++" and "--" provided for appending and deleting objects in a list.** The ability to append or delete one or more entries in a list of objects has been provided. Required only by advanced users. Details will be provided in the User Manual.

Changes in release 2015-47:

- **License changed to Apache 2.0.** The open-source software license for JaamSim has been changed from GPL3 to Apache 2.0. This change means that users can develop and sell their own commercial software packages based on JaamSim at no extra charge. The resulting software is not required to be open-source.
- **Working and Idle States for EntityDelay.** An EntityDelay is set to "Working" whenever one or more entities are undergoing this object's delay activity. This is the same logic as EntityConveyor.

Changes in release 2015-46:

- **Experimenter.** A series of simulation runs can be executed with changes to one or more inputs between runs. A new tab called "Multiple Runs" with suitable keywords has been added to Simulation to provide this capability. Model inputs can be changed between runs using the new outputs "RunNumber" and "RunIndex" which can be referenced in expressions. The output "RunIndex" is intended to allow the model to sweep over all the combinations of multiple inputs. However, this capability will not be available until expressions can reference an indexed array such as [Simulation].RunIndex(2). This feature is coming soon -- sometime in the next few weeks.
- **GlobalSubstreamSeed can accept an expression.** Multiple replications can be executed automatically by entering [Simulation].RunNumber for this keyword and specifying the number of runs to make using the keywords in the "Multiple Runs" tab.
- **Unit conversion bug in Output Viewer has been fixed.** Unit conversions for some vector outputs such as UnitsInUseTime for Resource were not being done correctly in the Output Viewer.
- **Warm-up period handled correctly.** Several bugs were fixed related to the InitializationDuration keyword. All the objects in the Process Flow palette now reset their statistics correctly at the end of the warm-up period.
- **Integer values are shown without decimal places in the Output Viewer.** Formerly, an integer such as "1" was displayed as "1.00000".

Changes in release 2015-45:

- **Generate entities from a schedule**. A new object "EventSchedule" has been added to the Basic Object palette that allows the user to enter a series of event times in either date or number format. These times are converted internally to inter-arrival times, allowing the EventSchedule object to be accepted by the EntityGenerator keyword "InterArrivalTime". A new inter-arrival time is returned each time the EventSchedule object is queried.
- **Generate a sequence of values**. A new object "ValueSequence" has been added to the Basic Objects palette that allows the user to enter a list of values. This object returns the next value in the sequence each time it is queried.

Changes in release 2015-44:

- **Orientation of a queued entity is restored when it leaves the Queue.** The Queue object temporarily alters the orientation of the queued entities so that they line up nicely. Formerly, the original orientation was not restored when the entity left the queue.
- **Deleting an entity causes its EntityLabel to be deleted**. Formerly, the EntityLabel was left behind and had to be deleted separately.
- **DowntimeEntity introduced**. This is the first step towards implementing maintenance and breakdowns for objects such as Server.

- **ExpressionEntity**. This object has been reinstated in JaamSim and has been modified to allow the user to specify the unit type for the expression being evaluated.

Changes in release 2015-43:

- **Attribute bug fixed.** The bug introduced in release 2015-42 has been fixed. The bug in question caused the Output Viewer to crash when an entity with an attribute is selected.

Changes in release 2015-42:

- **New output "StateTimes" has been added**. The StateTimes output reports the total time that the object has spent in each of its states.
- **Improved Output Viewer**. Outputs are shown in logical order and in the units selected by the Simulation keyword DisplayedUnits.
- **Improved output report**. New header information about the simulation run has been added, array outputs are spread over multiple lines, objects are listed alphabetically by class name and object name, and units selected by the DisplayedUnits keyword are used throughout.
- **Improved EntityLogger report**. Header information and the time for each log entry have been added.
- **Initialization bugs fixed**. Queue and Resource have been modified so that they work correctly with the InitializationDuration keyword for Simulation.
- **Improved description for Distribution output "Value"**. The circumstances that trigger a new sample being selected are now described correctly.

Changes in release 2015-41:

- **Stop/Pause condition**. A new keyword PauseCondition for Simulation allows the user to specify an expression that pauses the simulation run. The ExitAtPauseCondition keyword allows the run to be terminated if the PauseCondition expression returns true.
- **EntityLogger reports statistics for states that have not been used**. A new keyword DefaultStateList for SimEntity allows the user to specify a set of states that are always printed to the output report, even if the elapsed time for a state is zero.

Changes in release 2015-40:

- **Expression Logger can trace values.** ExpressionLogger can now record a log entry every time an expression changes its value. With this improvement, it can record log entries at regular intervals and/or when an object's state changes and/or when an expression's value changes.

Changes in release 2015-38:

- **Text object can display the value of an expression.** The OutputName keyword has been renamed to DataSource and generalized so that it can accept an expression as well as an Output. Inputs to DataSource are now given in the standard expression notation instead of as a list of output names. For example, instead of the old input Queue1 QueueLength, you now enter [Queue1].QueueLength. Both the keyword name change and the input format have been made backward compatible with older models. Lastly, a new keyword "UnitType" has been added. To maintain backwards compatibility, it has been made optional for the time being, however UnitType should always be specified for new models.
- **ExpressionLogger can display text values**. Expressions that return a text value can now be entered for the DataSource keyword.

- **ExpressionLogger has new keywords that restrict the portion of a simulation run to be logged**. New keywords StartTime, EndTime, and IncludeInitialization have been added to ExpressionLogger.
- **ExpressionLogger can record state changes.** The new keyword StateTraceList allows the user to specify a set of object to be traced. A record is added to the log at the exact time that any one of the objects changes state.
- **Property Viewer shows values for expressions and integer arrays**. Formerly, these properties were shown as object names, which was not much help to the programmer.

Changes in release 2015-35a:

- **Input Editor bug fixed.** A bug has been fixed that was introduced in release 2015-35, preventing the Input Editor from working correctly for many of the keywords labelled as "Required Inputs".
- **Resource bug fixed**. A bug has been fixed that caused model execution to stop under certain conditions when a Resource was released.

Changes in release 2015-35:

- **Just a few minor improvements this week.**

Changes in release 2015-34:

- **New InputValue object.** The InputValue object allows the user to enter a numerical value that can be used as an input to one or more other objects.
- **New choose function for expressions.** The function choose(i, E1, E2, E3, ... En) returns the value for the i-th entry in the list. The entries can be numbers or expressions. For example, the expression 'choose(2, 1[m], 2[m], 3[m])' will return the value 2[m].
- **New keyword for SignalThreshold to specify the initial state.**
- **EntityConveyor and EntityDelay respect their RelativeEntity input**. The entities now follow the polyline correctly when RelativeEntity has been specified.
- **Object Selector clean-up**. A number of changes were made to reduce clutter. The Simulation object is now shown without the palette and object type folders. Units and Views have been removed. Work is under way to allow the DisplayModel palette to be removed.
- **Input Editor and Output Viewer clean-up.** The Description field has been removed as both an input and an output. This input seemed like a good idea at the time, but we found that it was never used and created unnecessary clutter.
- **Position, Size, Alignment, and Orientation keywords hidden for polyline type objects.** For polyline type objects, these keywords are replaced in the Input Editor by the Points keyword.

Changes in release 2015-33:

- **States work correctly for duplicated entities.** The entities created by a Duplicate object are now initialized with the same state as the original entity.
- **Statistics object respects unit type**. A new UnitType keyword has been added to the Statistics object. Outputs for the Statistics object now use the correct unit.
- **JaamSim now works correctly for Windows users whose names include accented characters.**

Changes in release 2015-32:

- **Major re-work of polyline objects.** Polyline objects such EntityDelay and EntityConveyor now respect their RelativeEntity keyword and the global SnapToGrid setting. Its label now follows it correctly. The Center in View action in the right-click menu now works correctly.

Changes in release 2015-31:

- **InputBox bug fixed**. If is now safe to double-click on an InputBox before its TargetInput keyword has been set.
- **ShapeModels such as Circle and Rectangle are updated immediately**. Changes to the ShapeModel inputs FillColour, OutlineColour, Filled, and Bold are now shown immediately on any DisplayEntity that uses this DisplayModel.

Changes in release 2015-30:

- **InputBox object.** A text object that assigns its value to the keyword for a specified object. Allows a model input to be entered directly in the view window.

Changes in release 2015-29:

- **EntityLabel object.** A new "EntityLabel" object is used to label the entities in a model. It is similar to the usual Text object except that changing the displayed text changes the name of the object that it labels. This is now the preferred method for changing an object's name. An EntityLabel can only be created using the "Added Label" menu item in the right click (context) menu. Only one EntityLabel is allowed per object.
- **ExpressionLogger object.** A new "ExpressionLogger" object prints the value of one or more expressions at regular intervals. The output report is named <run name> - <ExpressionLogger name>.log and is saved to the report directory.

Changes in release 2015-28:

- **Edited text is highlighted correctly for an angled Text object.** The position of the highlighted text is shown correctly when the Orientation keyword has been set to a non-zero value.
- **"ScreenPointsModel" in the Display Models palette has been renamed as "PolylineModel"**. Linear objects such as EntityConveyor and EntityDelay use this type of display model. Should have no effect on existing models.
- **"DisplayModel" in the Display Models palette has been renamed as "ShapeModel"**. Flat objects that change colour such as ExpressionThreshold use this type of display model. If after loading your model you see an entry for "DisplayModel" in the Object Selector under the Display Models palette, then you will need to edit its configuration file to make the following change -- replace the text "Define DisplayModel" with "Define ShapeModel". This will be required for very few models. The old DisplayModel object will be available for a few more weeks before being deleted so that users have extra time in which to update their models.

Changes in release 2015-27:

- **Text objects can be edited in place using the mouse**. When editing a Text object in place, the mouse can be used to position the cursor and to highlight text. Double clicking selects an entire word.
- **Cut operation implemented for edited Text objects**. Highlighted text can be cut using the keyboard command Cntrl-X. The cut text is copied to the clipboard.

Changes in release 2015-26:

- **Entity movement using the arrow keys respects SnapToGrid.** Movements with the arrow keys now respect the SnapToGrid setting.
- **Text objects can be edited in place**. A Text object can be edited in the view window by double clicking on the Text object or by pressing F2 after it has been selected. Text can be entered, deleted, copied (control C), and pasted (control V) in the normal way. At present, only the arrow keys can be used to position the cursor. Mouse interactions with the edited text will come in the next few weeks.

Changes in release 2015-25:

- **Snap to Grid.** When Snap to Grid is set, a dragged object will automatically position itself on the nearest grid point. Snap to Grid can be turned on and off from a new entry in the Options menu. The snap grid's spacing is set using the Simulation keyword SnapGridSpacing under the GUI tab.
- **Moving an entity using the arrow keys.** The selected entity can now be moved in the x, y, and z directions using the arrow keys. The amount moved with each click is set by the Simulation keyword IncrementSize under the GUI tab.
- **Moving the camera using the arrow and WASD keys.** When the View window is selected, the arrow keys move the camera similar to the controls for a computer game. Following game conventions, the arrow keys are mirrored by the WASD keys.

Changes in release 2015-22:

- **Default random seed removed**. The default value for the RandomSeed keyword has been set to "None" and this input has been a required input. New models built by drag and drop are not affected because JaamSim automatically sets the input for this keyword to the next unused random seed value. However, some old models may rely on the default value which was set to zero. If this is the case for your model, simply load the model, click on Okay when the input error is detected, enter a suitable value for the RandomSeed keyword, and re-save your model.
- **Tool windows move to front**. When moving between separate applications in Windows, clicking on the JaamSim icon in the Task Bar brings all the selected JaamSim windows to front.
- **Closing the Save Changes dialog box is the same as "Cancel"**.

Changes in release 2015-21:

- **Huge queues**. Queues now work efficiently with huge numbers of waiting entities and huge numbers of Match values.

Changes in release 2015-20:

- **Required inputs**. Inputs for an object that must be provided by the user are now indicated in the Input Editor. All objects except those in the Calculation and Fluid Flow palettes have been programmed for this function.
- **Keyword pop-ups show an example for each type of input**. Keywords that can accept several types of inputs (e.g. a number, probability distribution, time series, etc.) now show an example of each type of input in their Input Editor pop-up.
- **Improved pop-ups for selected keywords**. Improved descriptions have been provided for GlobalSubstreamSeed, RandomSeed, AttributeAssignmentList, and RunDuration.
- **Better information on default values**. The Default column in the Input Editor now provides useful information for keywords whose default was previously shown as blank. Where no default is provided,

the text "None" is displayed. Where the default has some special meaning, the appropriate text (such as "All Views" or "This Entity") is displayed.

- **DiscreteDistributions with huge numbers of probabilities**. DiscreteDistribution can now sample efficiently from distributions with huge numbers of probabilities. The test case provided by a user had 80,000 probabilities to choose from.

Changes in release 2015-19:

- **Match keyword added to Server, etc**. The new keyword allows you to specify which entity should be removed next from the queue. For example, if the present value for the Server's Match keyword is 5, then it will take the first entity in the queue whose value for the queue's Match keyword is also 5. Both keywords can accept an expression. The Match keyword is implemented for all the Process Flow objects that use the WaitQueue keyword: Server, EntityGate, Seize, Pack, Unpack, AddTo, and RemoveFrom.

Changes in release 2015-18:

- **Re-written Object Selector**. The new Object Selector solves three problems: duplicated palette entries, presence of deleted entities, and an unresponsive GUI when there are large numbers of entities. On this last point, the new Object Selector can easily handle 10,000 entities on a typical computer. With even greater numbers of entities, your computer will run out of RAM eventually and slow down dramatically. At this point, the GUI will start to become unresponsive and JaamSim and will likely need to be shut down with the Windows Task Manager.
- **Entity names can be changed without pressing Enter.** When changing the name of an entity in the Object Selector, it is no longer necessary to press Enter. Clicking on anything else will accept the new name.
- **AttributeDefinitionList added to ExpressionThreshold.** This keyword has now been restored to ExpressionThreshold.
- **Various bugs fixed.** The NonStatExponentialDist object now returns a number with units of time. Error messages related to expressions have been restored.

Changes in release 2015-17:

- **Match parts to be assembled**. Keywords "Match" and "MatchRequired" have been added to the Queue and Assemble objects respectively. The Match keyword for Queue allow the user to enter an expression that is evaluated when each entity arrives at the Queue. When the MatchRequired keyword is set to TRUE for the Assemble object, the parts that are assembled are restricted to those with the same Match values.
- **Number of parts to assemble.** A new keyword "NumberRequired" has been added to Assemble. This keyword allows the user to specify how many entities should be removed from each queue to use in the assembly process.
- **Combine object.** A new object named "Combine" has been added to the Process Flow palette. This object provides the opposite of the Duplicate function, that is, it takes one entity each from a set of Queues and effectively combines them. The entity in the first Queue is passed on to the next object while those from the other Queues are destroyed. Each entity must have the same Match value.
- **New outputs for Queue.** Outputs for the waiting times, Match values, and Priority values have been provided for each entity in the queue.
- **New output provides the ObjectType.** An output has been added that gives the object type of the entity.
- **Last directory is remembered.** JaamSim now remembers the directory from which the last configuration file was loaded from or saved to. A big time saver when loading or saving configuration files.
- **Stop button renamed to Reset.** The Stop button has been renamed as the "Reset" button.

Changes in release 2015-16:

- **Region can be dragged and dropped**. The Region object has been given a graphic representation and can be dragged and dropped from the Model Builder.
- **Region and RelativeEntity keywords work together correctly**. Entities that have inputs for both the RelativeEntity and Region keywords are now positioned correctly.
- **VideoRecorder can be dragged and dropped.** The VideoRecorder object has been given a graphical representation and can be dragged and dropped from the Model Builder.
- **ProcessPosition keyword**. Specifies the position for the entity being processed by Server, EntityGate, Assemble, Pack, Unpack, AddTo, and RemoveFrom.
- **PositionOffset keyword for EntityContainer.** Specifies the positions for the entities carried by the EntityContainer.
- **File paths stay relative.** Inputs specifying a relative file path are no longer converted to an absolute file path when the model is saved.
- **Misc. bug fixes.** The two bugs noted by Jerry Levasseur related to the Text object have been fixed.
- **Misc. GUI improvements.** New dialogue box when a configuration file with errors is loaded.

Changes in release 2015-15:

- **Time varying random generator**. A new object named "NonStatExponentialDist" has been added to the Probability Distributions palette. When combined with a TimeSeries and an EntityGenerator, these objects create a stream of random arrivals with a time varying arrival rate. The correct "non-stationary Poisson process" algorithm is used.
- **Duplicate object copies the present values of the attributes.** An entity copied using the Duplicate object will have the same present values for its attributes as the original.
- **"Add Label" positions the label correctly when RelativeEntity is set.** Objects with a RelativeEntity can now use labels correctly.
- **Multiple entities can be chained using the RelativeEntity keyword**. Entity A can be relative to entity B, entity B can be relative to entity C, and so on without limit. Choices of RelativeEntity that would cause a circular reference are prevented automatically by the code.
- **New icon for the Stop button**. The Stop button has been replaced with a new round button with a "rewind" symbol.
- **New animations for the Run/Pause button**. The Run/Pause button now differentiates between the button pressed and button released states. The button also has a nice "roll-over" effect (the button brightens with the mouse is over it), but this is only visible on some of the Windows themes such as "Aero". Unfortunately, it is not supported by the "Windows Classic" theme.
- **Missing braces are trapped by TimeSeries keyword "Value"**. The code now ensures that the required opening and closing braces are included.
- **Modulo operator '%' can be used in expressions**. The expression '10 % 4' is equal to 2.
- **Missing single quotes are added automatically to inputs to ExpressionThreshold**. The "OpenCondition" keyword for ExpressionThreshold will now accept an expression that normally would be enclosed by single quotation marks (because it includes spaces).
- **LogNormalDistribution has been modified so that it respects units correctly**. The keywords "NormalMean" and "NormalStandardDeviation" now require dimensionless values. A new keyword "Scale" has been provided to multiply the output value by a constant with the desired units.

Changes in release 2015-14:

- **Units for default value**. Default values In the Input Editor are now shown in the preferred unit set by the user. For example, if the preferred unit for time is hours, the default value for RunDuration is '8760 h', instead of '3.1536E7 s'.
- **"PreferredUnit" keyword moved to Simulation**. The Unit keyword "PreferredUnit" has been moved to the Simulation object, where it is more accessible. It has been renamed "DisplayedUnits" and now accepts a list of all the preferred units in one place.
- **New Outputs for Process Flow objects**. New outputs "Working" and "Open" have been added to the relevant objects in the Process Flow palette to track the object's state.
- **"GlobalSubstreamSeed"** keyword moved to Simulation. The probability distribution keyword "GlobalSubstreamSeed" has been moved to the Simulation object where its global nature is more obvious.
- **TimeSeries and TimeSeriesThreshold**. Further improvements to TimeSeries and TimeSeriesThreshold have been made. These changes were needed to provide a sound foundation for the Non-Stationary Poisson generator that is being developed.
- **Progress Bar reset.** The progress bar is now reset to zero when the Stop/Reset button is clicked.
- **PauseTime in date/time format.** A date/time entered in the PauseTime field on the Control Panel will automatically be enclosed in single quote marks, if required.
- **TimeSeries and TimeSeriesThreshold inputs are cleared when unit type is changed.** Inputs to TimeSeries and TimeSeriesThreshold keyword that depend on the unit type input will be cleared if the unit type is changed.
- **"AttributeDefinitionList" keyword removed from most objects.** The "AttributeDefinitionList" keyword has been removed from all objects except those in the Process Flow palette.
- **Text and OverlayText objects display "Input Error" when there is a problem.** Previously, the screen went blank when there was a problem with the unit type or the format field.

Changes in release2015-13:

- **ExpressionThreshold bug**. The crasher bug reported by Jean-Christophe involving ExpressionThreshold has been fixed.
- **TimeSeries and TimeSeriesThreshold**. Internal improvements that increase execution speed for very large data sets.

Changes in release 2015-12:

- **New Control Panel.** To provide more room for the view window, the Control Panel has been re-designed so that everything fits into a single row of tools and outputs.
- **Different graphics for each state.** Many objects can change their appearance when they change state using the StateGraphics keyword.
- **New views use the most recent camera settings.** A new view window inherits the camera position from the current view window.
- **Improvements to the Model Builder.** Thumbnail icons have been improved to make them appear less pixelated. Unnecessary appearance of a horizontal slider has been suppressed.

Changes in release 2015-11:

- **RelativeEntity bugs**. The bugs involving RelativeEntity reported last week have been fixed. Server, Assemble, Pack, and Unpack now work correctly when RelativeEntity is set.

- **Excessive CPU usage**. The unnecessary CPU usage reported last week has been fixed.
- **Selected entity at start-up**. The entity shown in the Input Editor and Output Viewer is now the Simulation object instead of View1.
- **Default colours shown as names**. The default value for a colour input is now shown in the Input Editor by its colour name instead of its RGB values. For example, a default colour is now shown as { red } instead of { 255 0 0 }.
- **Remaining run time**. The remaining time in a simulation run shown in the Control Panel is now displayed in seconds, minutes, hours, days, or years depending on the magnitude of the value.
- **Options menu items for "Show Grid" and "Show Axes"**. New entries have been added to the Control Panel's Options menu to determine whether the grid on the x-y plane and the x-, y-, and z-axes should be displayed.
- **Speed Multiplier greyed out when Real Time is not selected**. The input box for Speed Multiplier in the Control Panel is shown in grey when the Real Time button is not depressed.
- **Control Panel layout**. Minor changes were made to the layout of the Control Panel. Tooltips are now provided for all the components of the Control Panel. More changes to the Control Panel are being planned.

Changes in release 2015-10:

- **New round button for play/pause**. Replaces the old square button with a round image super-imposed on it. This is the first step in a major rework of the Control Panel that is under way.
- **Branch object bug fix**. Branch now updates the NumberProcessed and ReleaseTime outputs correctly.
- **RelativeEntity bug fix**. EntityDelay, EntityConveyor, EntityContainer, and Queue were fixed so that entities with RelativeEntity set are positioned correctly.

Changes in release 2015-09:

- **Graph works with expressions**. The keywords DataSource and SecondaryDataSource can now accept expressions as inputs. New keywords UnitType and SecondaryUnitType are added to specify the units to be used for the primary and secondary y-axes. Graph is backwards compatible with the old format, e.g. { Entity1 Output1 } { Entity2 Output2 }, and for this format it is not necessary to specify the UnitType or SecondaryUnitType keywords. However, this will be an undocumented feature, and it would be best to convert these inputs to the new format: { [Entity1].Output1 } { [Entity2].Output2 } and set the UnitType input. You can also enter constants and any object that returns a number such as a TimeSeries, a ProbabilityDistribution, or a Calculation, as long as all the entries have the same unit type, e.g. { 1 m } { [Entity1].Output1 } { TimeSeries1 }.
- **Unnecessary prompts to save are suppressed**. A number of common interactions with the model will no longer trigger a prompt to save your model. Any changes will continue to be saved with the model, but you won't be prompted to do so. The interactions in question are setting real time mode, changing the speedup factor, setting a pause time, opening/closing the Model Builder, Input Editor, etc., opening or closing a view window, or panning, zooming, etc. within a view window.
- **SetGraphic object no longer changes the DisplayModel input**. The SetGraphic object has been modified so that it does not change the input for the DisplayModel keyword. Formerly, this would prevent the model graphics from resetting correctly on restart and would trigger a prompt to save your changes when the model is closed.

Changes in release 2015-08:

- DisplayEntity creation speed increased 50%. This is the bottleneck for many models -- you may see a significant increase in execution speed.
- SimEntity creation speed now similar to DisplayEntity. It was VERY slow in previous releases.
- Seize and Release can now accept an Expression for their NumberOfUnits keyword, allowing a variable number of units to be seized or released. See attached example. Note that the format for the NumberOfUnits input is now { 1 } { 1 } for two resources, whereas it was formerly just 1 1. For a single resource, the input can be either { 1 } or 1.
- Graph now accepts any number of entries for the LineColours, LineWidths, SecondaryLineColours, and SecondaryLineWidths keywords. They do not have to match the number of entries for DataSource and SecondaryDataSource anymore.
- Graph now accepts any number of entries for the XLinesColor and YLinesColor keywords. They do not have to match the number of entries for XLines and YLines anymore.
- The UnitType keyword for Probability Distributions and other objects can be reset from one unit type to another. Note that any other keywords whose unit type is determined by the UnitType input will be reset to their default values. Formerly, UnitType could only be set once -- you had to create a new object if you wanted a different unit type.

Changes in release 2015-06:

- Exceptions thrown during model runtime are now written to the log file, and not just the pop-up error window, makes it much easier to do a post-mortem.
- New faster way of copying entities, useful for process-flow models that generate a lot of entities from a prototype.
- Improved seeding of the random number generator cuts down on initialization time considerably when lots (> 1000) of random variables are used in the same model.

Changes in release 2015-04:

- The ContinuousDistribution now validates more up-front when parsing the CumulativeProbabilityList keyword, instead of validating it when the model is started. This lets Input errors be caught much earlier.
- The tooltips have had a cosmetic going-over to make them more consistent

Changes in release 2015-03:

- new keyword "TargetEntity" added to the SetGraphics object
- sending an entity directly to an Assemble object now causes a runtime error (they must be sent to the Assemble object's queues)
- sending an entity to an EntityGenerator now causes a runtime error (instead of failing quietly)
- improved tooltip popups for the Control Panel and Model Builder (now similar to the ones in Microsoft Office)

Changes in release 2015-02:

- The output "obj" for all objects that use queues (i.e. Server, Seize, Assemble, Pack, Unpack, AddTo, RemoveFrom, and EntityGate) is now the entity being processed by the object. Formerly, it was the entity that had been received most recently, and possibly sent to the queue.

- The output "NumberAdded" for the objects that use queues (see above) is now incremented when the object starts processing the entity. Formerly, it was incremented when the object first arrived at the object, and was possibly sent to the queue.
- The keyword "Choice" has been removed from Assemble. The entities to be assembled must now be sent directly to the appropriate queues. Entities sent to the Assemble object will be sent to the first queue in the WaitQueueList as a default. This change was necessary because the Choice keyword relied on the obj output to determine which queue an incoming entity should be sent to. This method does not work now that obj is the entity being processed, not the one that just arrived.
- A new keyword "FIFO" has been added to queue. If FIFO is TRUE (the default) then queues behave normally. If it is FALSE, then entities are queue in last-in-first-out (LIFO) order within each priority value.
- Simulation keyword 'SimulationTimeScale' has been replaced with 'TickLength' which allows you to directly specify the length of the internal simulation steps. The default has also been changed to be 1 microsecond.
- A bug has been fixed that forced the user to press the stop button when their model fails a validation check at startup, now you can just fix your inputs and hit play again without the extra button press to restart things.
- The default value that appeared for Unit's keyword 'ConversionFactorToSI' has been fixed in the EditBox, this was a cosmetic issue and had no effect on model results
- the Simulation functionality provided through the 'PauseTime' keyword now takes effect in all cases, including when the model is run in batch mode.
- improved icons have been provided for some types that did not provide a good preview image before in the Model Builder
- Lots of refactoring in the Process Flow objects, I'll let Harry expand on new functionality there
- For those people coding their own objects, many of the most common simulation/eventmanager methods have had their javadocs added or improved

Changes in release 2015-01:

- The deadlock issue reported earlier has been fixed, please report if you see any others.
- saved config files now save the unit types before other dependent keywords avoiding errors on reload
- additional validation has been added to some keywords where only a certain range of values is allowed
- EntityGenerator can create multiple entities per arrival time
- Queues can be shared between multiple Servers, Seize blocks, etc.
- Entities can be sent directly to a Queue instead of via a Server, Seize block, etc. The queue notifies the all its Servers, etc. whenever an entity is added to the queue.
- Queues can use a priority input to sort the entities (see attached example)
- Seize blocks now send all entities to its queue before processing
- Seize blocks now implement the OperatingThresholdList keyword
- keywords that accept a probability distribution or a time series now test on start-up that the minimum and maximum values for these sources are valid for the input. For example, the input for ServiceTime keyword for Server must be >= 0 s. A NormalDistribution will generate a validation error on start-up unless the input for its MinValue keyword is >= 0 s.

Changes in release 2014-52:

- Unit validation has been pushed into just about all places an expression can be evaluated

- Intrinsic functions for trig function (sin(), cos(), tan()) and inverse (asin(), acos(), atan(), atan2()) have been added to the expression engine
- Constants E() and PI() have been added
- ln(), log() and exp() have been added
- The GUI has been updated to only show the time remaining in a run when the model is running, and will show remaining seconds when the estimated remaining runtime is less than one minute.

Changes in release 2014-51:

- New keyword 'TraceState' that allows you to print a log file for each StateEntity containing a record of all state transitions in a flight-recorder kind of format. This is mainly useful as a debug tool allowing you to correlate state transitions across multiple entities.
- All of the KeyInput derived types had a bug where the no-key-specified value was being written over the default value rather than its own storage, which made clearing the input not reliable. This has been fixed for all Inputs of this type.

Changes in release 2014-48:

- improved validation of expressions more errors will now be caught at input time
- Improved error dialogs that identify the error location with greater clarity and consistency
- Pack and RemoveFrom objects now accept expressions for the NumberOfEntities keyword
- Random distributions that are drag-and-dropped will now be populated with a unique randomseed input to help those people building models interactively.
- default size/appearance for SimEntity, Queue and a few others have been changed
- The 'Process Flow' palette has been added to hold all the objects used to build models of that type as the basic objects palette was getting a bit crowded.

Changes in release 2014-45:

- JaamSim now requires Java 7 and above, Java6 compatibility has been dropped
- improvements to the speed entities can be generated, likely only noticeable on models generating very large numbers of entities.
- Utility windows will no longer appear as separate entries on the taskbar
- fixes to the View keywords size and position, now when an open window is adjusted in the editbox, it will respond immediately

Changes in release 2014-44:

- thread safety fixes for the ExpressionThreshold, this snuck in when we pushed unit aware evaluation into the expression engine
- better logging when a Texture fails to load in the renderer, helps to disambiguate out-of-memory from invalid texture data
- earlier discard of some 3d-asset data that will help a little bit with very large 3d models

Changes in release 2014-42:

- bugfix for expressionthreshold when the threshold can switch states during the same timestep and miss future updates.

Changes in release 2014-41:

- when selecting a polyline (like EntityDelay by default), the first point will be highlighted in blue, the middle points in green, and the end point in yellow, this was requested to give a better feel for the 'direction' of the line objects.
- small fix to videorecorder to ensure all updates to the model state are consistent before a frame is recorder, this was most noticeable with the camera sometimes being interpolated one frame behind the rest of the model.
- another fix to VideoRecorder to clear old video files before recording over them with a new capture.
- fix a hard to hit transparency bug with ImageModel
- Keywords are now looked up in a case-sensitive manner to match the rest of input parsing, a fallback is currently provided that will warn if the case-insensitive lookup is used, this will become a hard error in a future release.
- A new keyword 'PauseTime' has been added to simulation that makes it easy to have the model pause at a certain time, useful when debugging your models.
- Tooltips on some of the GUI elements have been improved
- Animation has been made optional for the EntityDelay object, this can speed up models where the interpolation of position is not wanted.

Changes in release 2014-39:

- Weibull distribution now takes a Location keyword which allows the returned values to be shifted
- Some fixes to the Collada texture loading code appropriately clamping/wrapping at boundaries
- Dropdowns now provided with a list of valid options for TimeSeries and SampleInputs
- For people programming directly in jaamsim, the duplicative ConditionalHandle class has been folded in to a single EventHandle class, updates to the programming manual will come out over the next few weeks.

Changes in release 2014-37:

- new icon for the Duplicate object
- default file name set for 'Save As' and 'Export 3D objects'
- Explicit .doubleValue() calls added to avoid build errors for people using the (strict) Eclipse Luna compiler
- Reduction of memory use for very high event-rate models, machines with low memory bandwidth may see speed improvements.
- Another increase in JaamSim's event processing speed! The time to schedule and process an event has been reduced to 280 clock cycles -- a 14% increase over last week.

Changes in release 2014-36:

- OSX users can now use Java 7 or 8 VMs to run Jaamsim, the graphical compatibility issue has been solved
- Jaamsim now runs on all platforms using Java 6, 7 or 8.
- a new Duplicate object has been added to the Basic Objects
- another big increase in execution speed.

Changes in release 2014-35:

- bugfix for ExpressionThreshold not being evaluated on repeated calls at the same time
- expressions now properly return errors in all cases, particularly when an entity name cannot be found.

- expressions have seen considerable performance enhancements at runtime
- large speedups to models using the ExpressionThreshold object

Changes in release 2014-34:

- a bug fix for repeated pressing the play/pause button in rapid succession could provoke an error in the model state
- significant speedup for expressionthreshold objects, models with many of these objects should run much faster

Changes in release 2014-33:

- New Pack and Unpack objects to group objects being passed around, example to follow
- Large amount of internal re-organization to remove old namespaces from the code, no user impact

Changes in release 2014-32:

- properly return an error when an included cfg file is unreadable, or not found (this was only returning a warning previously)
- update the internal version of JOGL to 2.2.0, if you see any 3D graphics oddities, please let me know!!

Changes in release 2014-30:

- the negative duration wait in EntityConveyor has been fixed
- an ExpressionEntity has been added to allow experimentation with the expression support
- Units are now being explicitly tracked in expressions, validation of correct units/types of expressions will follow over the coming weeks.

Changes in release 2014-29:

- When echoing out input being read from a file, fix missing linefeeds so the error message is actually readable
- Entities in a queue are positioned correctly when a Queue is relative to another object

Changes in release 2014-28:

- an implementation of the Beta distribution has been added to the list of ProbabilityDistributions
- A new EntityLogger has been added which allows log files to be captured during the run.
- StateEntities now report and store their state list alphabetically

Changes in release 2014-27:

- All linked components now track what state they are in and can produce statistics for the time spent
- objects being passed from component to component can also have a state set
- bug fixed when dragging points in a polyline (like EntityConveyor) when the object is in a Region
- bug fixed for ArrowHead size not updating consistently from input changes in the editor
- graphicbox no longer sets a zero size on an object if an imagemodel is set dynamically
- extra validation for the Size keyword no longer accepts negative values

Changes in release 2014-26:

- fixed the save as functionality to output defined attributes first, which is required if later inputs use them in expressions

- Program no longer goes into an infinite loop when the only OpenGL version found is too old to be used (we deadlocked trying to show an error message, how embarrassing ;-)) Now the error window will come up and inform you of the problem.
- When very large input is loaded into jaamsim, it is not echoed out which was build a very large string internally

Changes in release 2014-25:

- A new StateEntity class has been added which allows tracking time spent in states, only Threshold has been updated to use this so far, the rest of the basic entity objects will be converted shortly
- boolean outputs are now correctly coerced to double in expressions allowing expressions mixing double/Boolean
- VideoRecorder had a corner case fixed that could cause blacked-out frames for some camera settings while looking just fine in interactive windows.

Changes in release 2014-24:

- New ReportGenerator and Statistics classes to control the collection and generation of run statistics to report files, an example of how to use this is forthcoming
- legacy classes from the Java5 conversion have been dropped (StringVector and Vector)
- all parsing of inputs is now zero-copy using the KeywordIndex object
- EventHandles have completely displaced the scheduleSingleProcess API
- further javadoc cleanup to some of the basic event scheduling APIs

Changes in release 2014-23:

- A new Assemble object has been introduced allowing multiple objects to be combined as they are passed through the basic objects system, see the keyword help for details
- a new SignalThreshold type has been created which allows Thresholds to be controlled using expressions
- significant internal refactoring of the Input parsing will allow the removal of a few extra copies of the input data
- the Threshold class and subclasses has been significantly simplified and documented
- continuing to remove use of the legacy Vector and StringVector class that should be gone in the next version or two. These were transitional objects form the Java 5 days...time to let them go

Changes in release 2014-22:

- new ExpressionThreshold object which will allow user-specified expressions to shut off the basic objects based on a condition
- finally removed some legacy internal functions, getting towards a pretty clean core set of code

Changes in release 2014-21:

- small fixes to operator precedence in the expression parser
- add an Assign object to set the user-defined attributes using expressions
- fix to a few race conditions in the renderer that could lead to dropped frames (in particular, the first frame could end up with a stale camera position)
- removal of the boolean math objects that can be represented directly using expressions
- fixup for an infinite loop that snuck in with the new expression parsing code when showing the expressions in the Input Editor.

Changes in release 2014-20:

- Internal optimization on Input parsing should make parsing large files much faster and use much less memory
- New Red-black tree for managing the future event list makes us scale much more nicely with thousands of future events. I've stressed tested it out to 1 million events without hitting any real walls
- changes to real time and the real time factor now get saved to the input file
- initial work to add user-definable attributes to any entity see the AttributeDefinitionList keyword for details
- small optimizations to generating entities using an EntityGenerator
- use a hashmap in EntityDelay to avoid linear search which can be painfully slow if many entities are delayed concurrently

Changes in release 2014-19:

- bugfix for killEvents leaking Process objects, code was functionally correct, but could out of memory eventually, introduced in 2014-18
- performance work in input parsing, much less garbage being generated on large inputs, which should also help models that generate a lot of Entities
- comma is no longer accepted as a separator in input files, only space and tab are accepted
- Initial commit of infrastructure to parse and evaluate basic user-defined expressions, full availability coming soon as we hook it into the existing objects
- The Process class has become an internal implementation class and is no longer visible, all functionality has been consolidated in the EventManager class
- new APIs for interrupting/killing events using an EventHandle or ConditionalHandle allows the user to explicitly mark scheduleProcess/simWait/waitUntil calls as being interruptible/killable.

Changes in release 2014-18:

- when running purely event-based models, it was possible to hold off gui updates nearly indefinitely, drop some locking to ensure the UI does not get deadlocked
- It was found that exposing direct access to the Process class made it easy to introduce hard to debug modelling errors, a new API taking explicit EventHandles has been implemented and a refactoring is in progress to hide the Process class as an internal implementation detail. Updates to the programming manual are forthcoming.
- The above API change also allows for significantly better performing interrupt/kill event implementation which should be complete for the next release.
- The EntityDelay class has also had some scalability work done when delaying lots of entities concurrently.

Changes in release 2014-17:

- Thresholds have a new Output for their open/closed state
- work to provide dynamic text BooleanIndicators
- prelim work moving towards at least microsecond precision in most time-related inputs
- OpenGL fix causing a warning on newer machines re: glLineWidth

Changes in release 2014-16:

- fully re-entrant eventmanager for event based models can be multiple times faster
- experimental cross-platform jar file (windows/linux/macosx)

- arena-like attribute assignment code is a work in progress
- subwindows now try to follow the GUIframe's state to get out of your way

Changes in release 2014-12:

- preliminary MacOSX support for 3D graphics, all other functionality already works
- more consistent use of the filechooser for interactive loading of config files, or now, graphics assets
- fixes to passing file names on the command line in batch-run mode
- default the Log Viewer to a more reasonable size